



# **PicoScope 6**

## **PC Oscilloscope Software**

User's Guide



# Table of Contents

1 Welcome .....	1
2 PicoScope 6 overview .....	2
3 Introduction .....	3
1 Legal statement .....	3
2 Upgrades .....	3
3 Trade marks .....	4
4 Contact information .....	4
5 How to use this manual .....	4
6 System requirements .....	5
4 Using PicoScope for the first time .....	6
5 PicoScope and oscilloscope primer .....	7
1 Oscilloscope basics .....	7
2 PC Oscilloscope basics .....	8
3 PicoScope basics .....	8
1 Capture modes .....	9
2 How do capture modes work with views? .....	10
4 PicoScope window .....	11
5 Scope view .....	12
6 Overvoltage indicator .....	13
7 MSO view .....	14
1 Digital view .....	15
2 Digital context menu .....	16
8 XY view .....	17
9 Trigger marker .....	18
10 Post-trigger arrow .....	18
11 Spectrum view .....	19
12 Persistence mode .....	20
13 Measurements table .....	21
14 Pointer tool tip .....	22
15 Signal rulers .....	23
16 Time rulers .....	24
17 Phase (rotation) rulers .....	25
18 Ruler settings .....	27
19 Ruler legend .....	28
20 Frequency legend .....	28
21 Properties sheet .....	30
22 Custom probes .....	31
23 Math channels .....	32
24 Reference waveforms .....	33
25 Serial decoding .....	34

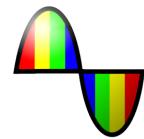
<b>26 Mask limit testing</b> .....	35
<b>27 Alarms</b> .....	36
<b>28 Buffer Overview</b> .....	37
<b>6 Menus</b> .....	38
<b>1 File menu</b> .....	39
<b>1 Save As dialog</b> .....	40
<b>2 Startup Settings menu</b> .....	46
<b>3 Waveform Library Browser</b> .....	47
<b>2 Edit menu</b> .....	48
<b>1 Notes</b> .....	48
<b>2 Channel Labels (PicoScope Automotive only)</b> .....	49
<b>3 Details dialog (PicoScope Automotive only)</b> .....	50
<b>3 Views menu</b> .....	51
<b>1 Custom grid layout dialog</b> .....	52
<b>4 Measurements menu</b> .....	53
<b>1 Add / Edit Measurement dialog</b> .....	54
<b>2 Advanced measurement settings</b> .....	55
<b>5 Tools menu</b> .....	57
<b>1 Custom Probes dialog</b> .....	58
<b>2 Math Channels dialog</b> .....	72
<b>3 Reference Waveforms dialog</b> .....	83
<b>4 Serial Decoding dialog</b> .....	85
<b>5 Alarms dialog</b> .....	92
<b>6 Masks menu</b> .....	94
<b>7 Macro Recorder</b> .....	97
<b>8 Preferences dialog</b> .....	98
<b>6 Help menu</b> .....	109
<b>7 Automotive menu (PicoScope Automotive only)</b> .....	110
<b>8 Connect Device dialog</b> .....	111
<b>9 Converting files in Windows Explorer</b> .....	112
<b>7 Toolbars and buttons</b> .....	114
<b>1 Advanced Options toolbar</b> .....	115
<b>2 ConnectDetect</b> .....	115
<b>3 Channels toolbar</b> .....	116
<b>1 Channel Options menu</b> .....	117
<b>2 Digital Inputs button</b> .....	123
<b>4 PicoLog 1000 Series Channels toolbar</b> .....	125
<b>1 PicoLog 1000 Series Digital Outputs control</b> .....	126
<b>5 USB DrDAQ Channels toolbar</b> .....	127
<b>1 USB DrDAQ RGB LED control</b> .....	128
<b>2 USB DrDAQ Digital Outputs control</b> .....	129
<b>6 Capture Setup toolbar</b> .....	130
<b>1 Spectrum Options dialog</b> .....	132
<b>2 Persistence Options dialog</b> .....	134
<b>7 Buffer Navigation toolbar</b> .....	136
<b>8 Measurements toolbar</b> .....	137
<b>9 Signal Generator button</b> .....	138
<b>1 Signal Generator dialog (PicoScope devices)</b> .....	138

<b>2 Signal Generator dialog (USB DrDAQ)</b> .....	141
<b>3 Arbitrary waveform files</b> .....	142
<b>4 Arbitrary Waveform Generator window</b> .....	143
<b>5 Demo Signals menu</b> .....	146
<b>6 Demo Signals dialog</b> .....	147
<b>10 Start / Stop toolbar</b> .....	148
<b>11 Triggering toolbar</b> .....	149
<b>1 Advanced Triggering dialog</b> .....	151
<b>2 Advanced trigger types</b> .....	152
<b>12 Zooming and Scrolling toolbar</b> .....	158
<b>1 Zoom Overview</b> .....	159
<b>8 How to...</b> .....	160
<b>1 How to change to a different device</b> .....	160
<b>2 How to use rulers to measure a signal</b> .....	161
<b>3 How to measure a time difference</b> .....	162
<b>4 How to move a view</b> .....	163
<b>5 How to scale and offset a signal</b> .....	164
<b>6 How to set up the spectrum view</b> .....	166
<b>7 How to find a glitch using persistence mode</b> .....	167
<b>8 How to set up a Mask Limit Test</b> .....	171
<b>9 How to save on trigger</b> .....	174
<b>10 How to create a link file</b> .....	178
<b>9 Reference</b> .....	180
<b>1 Measurement types</b> .....	180
<b>1 Scope measurements</b> .....	180
<b>2 Spectrum measurements</b> .....	181
<b>2 Signal generator waveform types</b> .....	183
<b>3 Spectrum window functions</b> .....	184
<b>4 Trigger timing (part 1)</b> .....	185
<b>5 Trigger timing (part 2)</b> .....	186
<b>6 Serial protocols</b> .....	187
<b>1 RS232/UART protocol</b> .....	188
<b>2 SPI protocol</b> .....	189
<b>3 I<sup>2</sup>C protocol</b> .....	190
<b>4 I<sup>2</sup>S protocol</b> .....	191
<b>5 CAN protocol</b> .....	191
<b>6 LIN protocol</b> .....	194
<b>7 FlexRay protocol</b> .....	196
<b>7 Device feature table</b> .....	198
<b>8 Command-line syntax</b> .....	199
<b>9 Application Error dialog</b> .....	202
<b>10 Flexible power</b> .....	203
<b>11 Glossary</b> .....	205
<b>Index</b> .....	209



# 1 Welcome

Welcome to **PicoScope 6**, the PC Oscilloscope software from Pico Technology.



With a scope device from Pico Technology, [PicoScope](#) turns your PC into a powerful [PC Oscilloscope](#) with all the features and performance of a benchtop [oscilloscope](#) at a fraction of the cost.

- [How to use this manual](#)
- [What's new in this version?](#)
- [Using PicoScope for the first time](#)

## 2 PicoScope 6 overview

**PicoScope 6** is Pico Technology's software for PC Oscilloscopes.

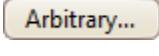
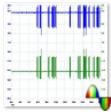
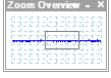
### ● Higher performance

- Faster capture rates, making it easier to see fast-moving signals
- Faster data processing
- Better support for the latest PicoScope USB oscilloscopes

### ● Improved usability and appearance

- Clearer graphics and text
- Tool tips and help messages to explain all features
- Easy point-and-click tools for panning and zooming

### ● New features

	The latest Windows .NET technology enabling us to deliver updates more quickly		Multiple <a href="#">views</a> of the same data, with individual zoom and pan settings for each view
	<a href="#">Custom probes</a> manager to make it easy for you to use your own probes and sensors with PicoScope		<a href="#">Advanced triggering conditions</a> including pulse, window and logic
	<a href="#">Properties sheet</a> displaying all settings at a glance		<a href="#">Spectrum mode</a> with a fully optimized spectrum analyzer
	Per-channel <a href="#">lowpass filtering</a>		<a href="#">Math channels</a> for creating mathematical functions of input channels
	<a href="#">Reference waveforms</a> for storing copies of input channels		<a href="#">Arbitrary waveform designer</a> for oscilloscopes with a built-in arbitrary waveform generator
	<a href="#">Rapid trigger mode</a> to capture a sequence of waveforms with the minimum possible <a href="#">dead time</a>		<a href="#">Windows Explorer integration</a> to show files as pictures and convert to other formats
	<a href="#">Command-line options</a> for converting files		<a href="#">Zoom overview</a> for rapidly adjusting the zoom to show any part of the waveform
	<a href="#">Serial decoding</a> for RS232, I <sup>2</sup> C and other formats, in real time		<a href="#">Mask limit testing</a> to show when a signal goes out of bounds
	<a href="#">Buffer overview</a> for searching the waveform buffer		<a href="#">Alarms</a> to alert you when a specified event occurs

## 3 Introduction

**PicoScope 6** is a comprehensive software application for Pico Technology oscilloscopes. Used with a PicoScope hardware device, it creates an oscilloscope and spectrum analyzer on your PC.

PicoScope 6 supports the devices listed in the [Device feature table](#). It runs on any computer with Windows XP SP3 through to Windows 8. (See [System requirements](#) for further recommendations.)

### How to use PicoScope 6

- ➊ Getting started: see [using PicoScope for the first time](#), and PicoScope's [Features](#).
- ➋ For further information: see descriptions of [Menus](#) and [Toolbars](#), and the [Reference](#) section.
- ➌ For step-by-step tutorials, see the "[How to](#)" section.

### 3.1 Legal statement

**Grant of license.** The material contained in this release is licensed, not sold. Pico Technology Limited ('Pico') grants a license to the person who installs this software, subject to the conditions listed below.

**Access.** The licensee agrees to allow access to this software only to persons who have been informed of and agree to abide by these conditions.

**Usage.** The software in this release is for use only with Pico products or with data collected using Pico products.

**Copyright.** Pico claims the copyright of, and retains the rights to, all material (software, documents etc.) contained in this release.

**Liability.** Pico and its agents shall not be liable for any loss or damage, howsoever caused, related to the use of Pico Technology equipment or software, unless excluded by statute.

**Fitness for purpose.** No two applications are the same, so Pico cannot guarantee that its equipment or software is suitable for a given application. It is therefore the user's responsibility to ensure that the product is suitable for the user's application.

**Mission-critical applications.** Because the software runs on a computer that may be running other software products, and may be subject to interference from these other products, this license specifically excludes usage in 'mission-critical' applications, for example life-support systems.

**Viruses.** This software was continuously monitored for viruses during production. However, the user is responsible for virus checking the software once it is installed.

**Support.** No software is ever error-free, but if you are dissatisfied with the performance of this software, please contact our technical support staff.

### 3.2 Upgrades

We provide upgrades, free of charge, from our web site at [www.picotech.com](http://www.picotech.com). We reserve the right to charge for updates or replacements sent out on physical media.

### 3.3 Trade marks

*Windows* is a registered trade mark of Microsoft Corporation. *Pico Technology*, *PicoScope* and *PicoLog* are internationally registered trade marks.

### 3.4 Contact information

**Address:**

Pico Technology  
James House  
Colmworth Business Park  
ST. NEOTS  
Cambridgeshire  
PE19 8YP  
United Kingdom

**Phone:**

+44 (0) 1480 396395

**Fax:**

+44 (0) 1480 396296

**Office hours:**

09:00 to 17:00 Mon-Fri

**Technical support email:**

[support@picotech.com](mailto:support@picotech.com)

**Sales email:**

[sales@picotech.com](mailto:sales@picotech.com)

**Web site:**

[www.picotech.com](http://www.picotech.com)

### 3.5 How to use this manual

If you are using a PDF viewer to read this manual, you can turn the pages of the manual as if it were a book, using the **Back** and **Forward** buttons in your viewer. These buttons should look something like this:



or



**Back**



or



**Forward**

You can also print the entire manual for reading away from your computer. Look for a **Print** button similar to this:



**Print**

For your first introduction to PicoScope, we suggest that you start with these topics:

- [Using PicoScope for the first time](#)
- [Oscilloscope basics](#)
- [PC Oscilloscope basics](#)
- [PicoScope basics](#)

### 3.6 System requirements

To ensure that PicoScope operates correctly, you must have a computer with at least the minimum system requirements to run your Windows operating system, which must be one of the versions listed in the following table. The performance of the oscilloscope will be better with a more powerful PC, and will benefit from a multi-core processor.

Item	Minimum specification	Recommended specification
<b>Operating system</b>	Windows XP SP3, Windows Vista, Windows 7 or Windows 8 32-bit or 64-bit edition Not Windows RT	
<b>Processor</b>	300 MHz	1 GHz
<b>Memory</b>	256 MB	512 MB
<b>Free disk space*</b>	1 GB	2 GB
<b>Ports</b>	USB 2.0 port	USB 2.0 port (USB 2.0 oscilloscopes) USB 3.0 port (USB 3.0 oscilloscopes)

\* The PicoScope software does not use all the disk space specified in the table. The free space is required to make Windows run efficiently.

## 4 Using PicoScope for the first time

We have designed PicoScope to be as easy as possible to use, even for newcomers to oscilloscopes. Once you have followed the introductory steps listed below, you will soon be on your way to becoming a PicoScope expert.

1.  Install the software. Load the CD-ROM that is included with your scope device, then click the "Install Software" link and follow the on-screen instructions.
2.  Plug in your scope device. Windows will recognise it and prepare your computer to work with it. Wait until Windows tells you that the device is ready to use.
3.  Click the new PicoScope icon on your Windows desktop.
4.  PicoScope will detect your scope device and prepare to display a waveform. The green **Start** button will be highlighted to show that PicoScope is ready.
5.  Connect a signal to one of the scope device's input channels and see your first waveform! To learn more about using PicoScope, please read the [PicoScope Primer](#).

### Problems?

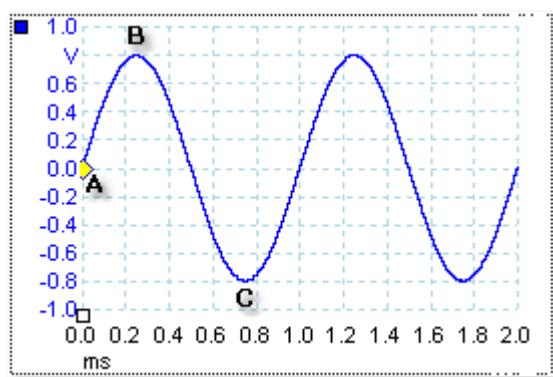
Help is at hand! Our technical support staff are always ready to answer your telephone call during office hours (see our [Contact Details](#)). At other times, you can leave a message on our [support forum](#) or [send us an email](#).

## 5 PicoScope and oscilloscope primer

This chapter explains the fundamental concepts that you will need to know before working with the PicoScope software. If you have used an oscilloscope before, then most of these ideas will be familiar to you. You can skip the [Oscilloscope basics](#) section and go straight to the [PicoScope-specific information](#). If you are new to oscilloscopes, please take a few minutes to read at least the [Oscilloscope basics](#) and [PicoScope basics](#) topics.

### 5.1 Oscilloscope basics

An **oscilloscope** is a measuring instrument that displays a graph of voltage against time. For example, the picture below shows a typical display on an oscilloscope screen when a varying voltage is connected to one of its input channels.



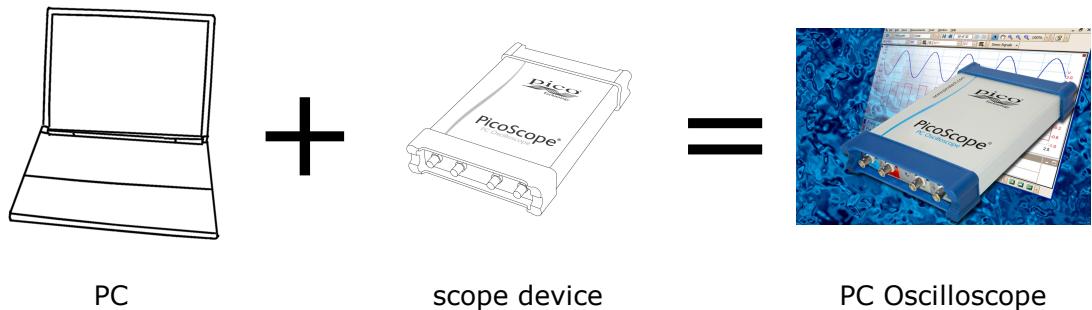
Oscilloscope displays are always read from left to right. The voltage-time characteristic of the signal is drawn as a line called the **trace**. In this example, the trace is blue and begins at point **A**. If you look to the left of this point, you will see the number "0.0" on the voltage [axis](#), which tells you that the voltage is 0.0 V (volts). If you look below point **A**, you will see another number "0.0", this time on the time axis, which tells you that the time is 0.0 ms (milliseconds) at this point.

At point **B**, 0.25 milliseconds later, the voltage has risen to a positive peak of 0.8 volts. At point **C**, 0.75 milliseconds after the start, the voltage has dropped to a negative peak of -0.8 volts. After 1 millisecond, the voltage has risen back to 0.0 volts and a new cycle is about to begin. This type of signal is called a sine wave, and is one of a limitless range of signal types that you will encounter.

Most oscilloscopes allow you to adjust the vertical and horizontal scales of the display. The vertical scale is called the **voltage range** (in this example at least, although scales in other units, such as milliamperes, are possible). The horizontal scale is called the **timebase** and is measured in units of time - in this example, thousandths of a second.

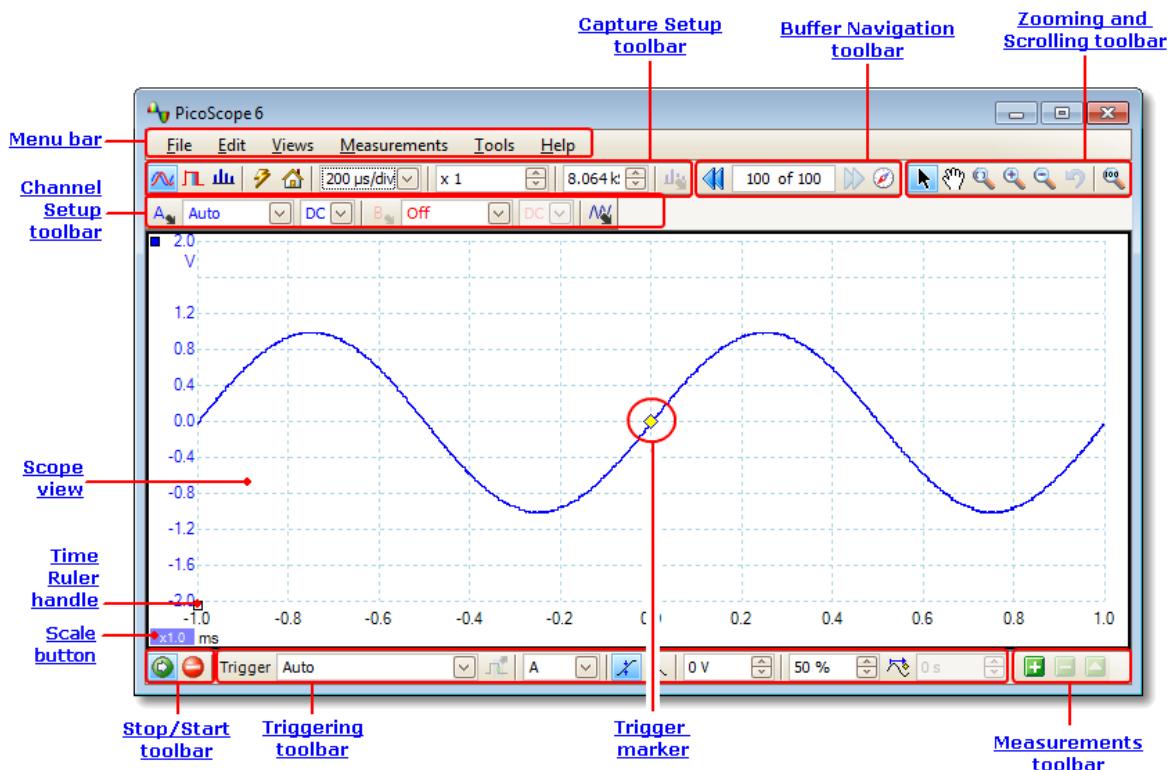
## 5.2 PC Oscilloscope basics

A **PC Oscilloscope** is a measuring instrument that consists of a hardware scope device and an oscilloscope program running on a PC. Oscilloscopes were originally stand-alone instruments with no signal processing or measuring abilities, and with storage only available as an expensive extra. Later oscilloscopes began to use new digital technology to introduce more functions, but they remained highly specialised and expensive instruments. **PC Oscilloscopes** are the latest step in the evolution of oscilloscopes, combining the measuring power of Pico Technology's scope devices with the convenience of the PC that's already on your desk.



## 5.3 PicoScope basics

PicoScope can produce a simple display such as the example in the [Oscilloscope basics](#) topic, but it also has many advanced features. The screen shot below shows the PicoScope window. Click on any of the underlined labels to learn more. See [PicoScope window](#) for an explanation of these important concepts.



**Note:** Other buttons may appear in the PicoScope main window depending on the capabilities of the oscilloscope that is connected, and on the settings applied to the PicoScope program.

### 5.3.1 Capture modes

PicoScope can operate in three capture modes: **scope mode**, **spectrum mode** and **persistence mode**. The mode is selected by buttons in the [Capture Setup Toolbar](#).



- In **scope mode**, PicoScope displays a main [scope view](#), optimises its settings for use as a PC Oscilloscope, and allows you to directly set the capture time. You can still display one or more secondary spectrum views.
- In **spectrum mode**, PicoScope displays a main [spectrum view](#), optimises its settings for spectrum analysis, and allows you to directly set the frequency range in a similar way to a dedicated spectrum analyzer. You can still display one or more secondary scope views.
- In **persistence mode**, PicoScope displays a single, modified scope view in which old waveforms remain on the screen in faded colors while new waveforms are drawn in brighter colors. See also: [How to find a glitch using persistence mode](#) and the [Persistence Options dialog](#).

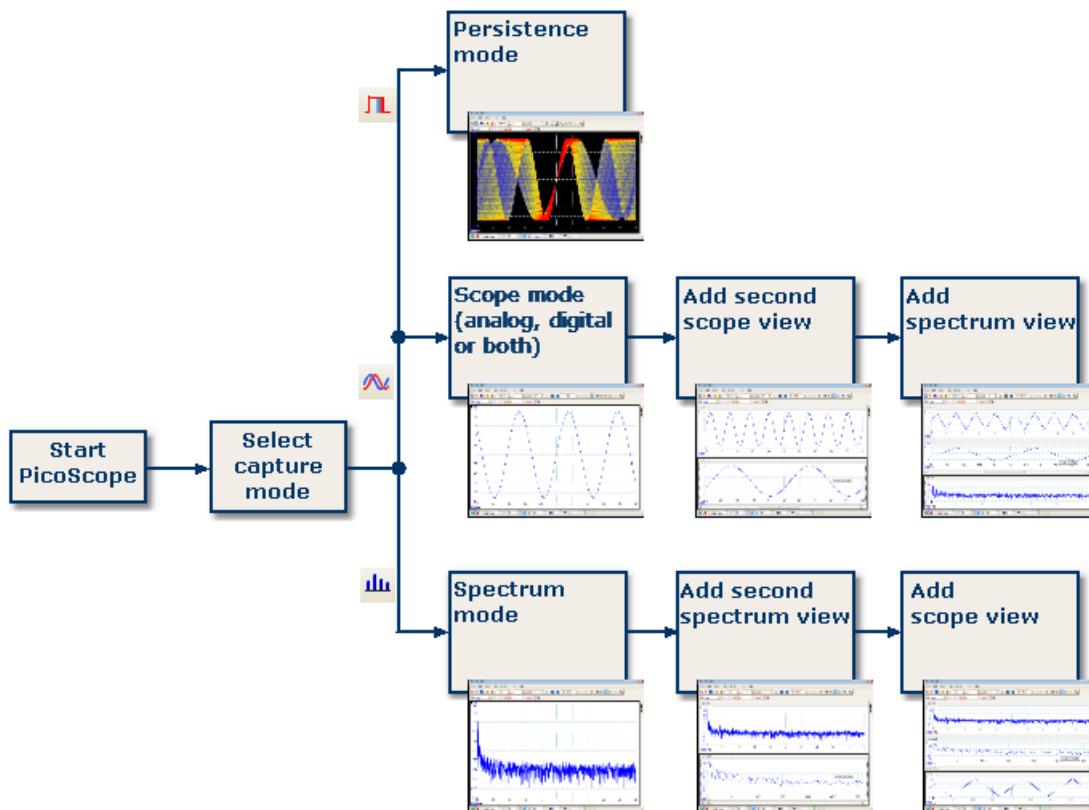
When you [save waveforms and settings](#), PicoScope only saves data for the mode that is currently in use. If you wish to save settings for both capture modes, then you need to switch to the other mode and save your settings again.

See also: [How do capture modes work with views?](#)

### 5.3.2 How do capture modes work with views?

The [capture mode](#) tells PicoScope whether you are mainly interested in viewing waveforms ([scope mode](#)) or frequency plots ([spectrum mode](#)). When you select a capture mode, PicoScope sets up the hardware appropriately and then shows you a [view](#) that matches the capture mode (a [scope view](#) if you selected scope mode or [persistence mode](#), or a [spectrum view](#) if you selected spectrum mode). The rest of this section does not apply in persistence mode, which allows only a single view.

Once PicoScope has shown you the first view, you can, if you wish, add more scope or spectrum views, regardless of the capture mode you are in. You can add and remove as many extra views as you wish, as long as one view remains that matches the capture mode.

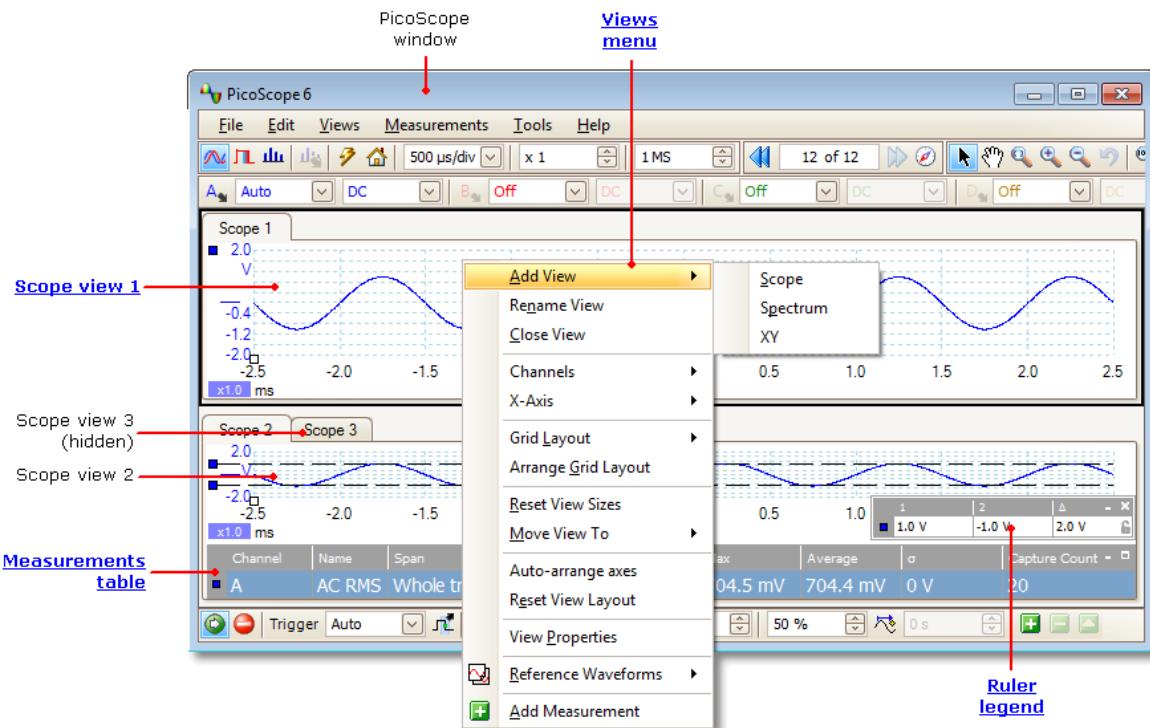


**Examples showing how you might select the capture mode and open additional views in PicoScope. Top: persistence mode (one view only). Middle: scope mode. Bottom: spectrum mode.**

When using a secondary view type (a spectrum view in scope mode, or a scope view in spectrum mode), you may see the data compressed horizontally rather than displayed neatly as in a primary view. You can usually overcome this by using the zoom tools.

## 5.4 PicoScope window

The **PicoScope window** shows a block of data captured from the [scope device](#). When you first open PicoScope it contains one [scope view](#), but you can add more views by clicking **Add view** in the [Views menu](#). The screen shot below shows all the main features of the PicoScope window. Click on the underlined labels for more information.



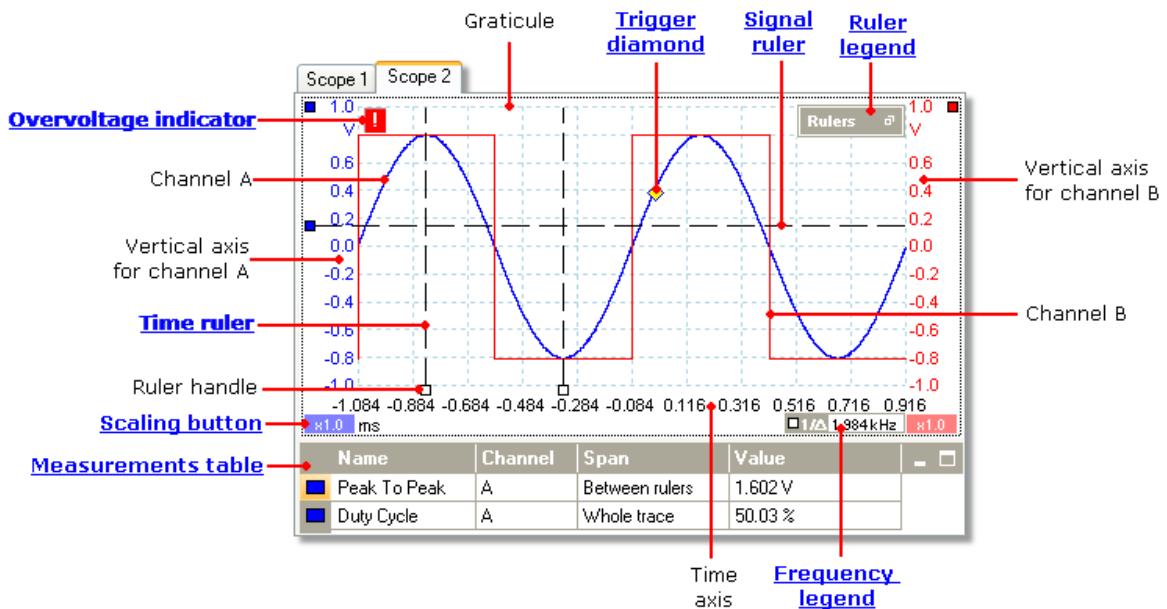
### To arrange the views within the PicoScope window

If the PicoScope window contains more than one [view](#), PicoScope arranges them in a grid. This is arranged automatically, but you can customize it if you wish. Each rectangular space in the grid is called a [viewport](#). You can move a [view](#) to a different [viewport](#) by dragging its name tab ([show me](#)), but you cannot move it outside the PicoScope window. You can also put more than one view in a [viewport](#), by dragging a view and dropping it on top of another.

For further options, right-click on a view to obtain the [View menu](#), or select **View** from the [Menu bar](#), then select one of the menu options to arrange the views.

## 5.5 Scope view

A **scope view** shows the data captured from the scope as a graph of signal amplitude against time. (See [Oscilloscope basics](#) for more on these concepts.) PicoScope opens with a single view, but you can add more views by using the [views menu](#). Similar to the screen of a conventional oscilloscope, a scope view shows you one or more waveforms with a common horizontal time axis, with signal level shown on one or more vertical axes. Each view can have as many waveforms as the scope device has channels. Click on one of the labels below to learn more about a feature.



Scope views are available regardless of which mode - [scope mode](#) or [spectrum mode](#) - is active.

## 5.6 Overvoltage indicator

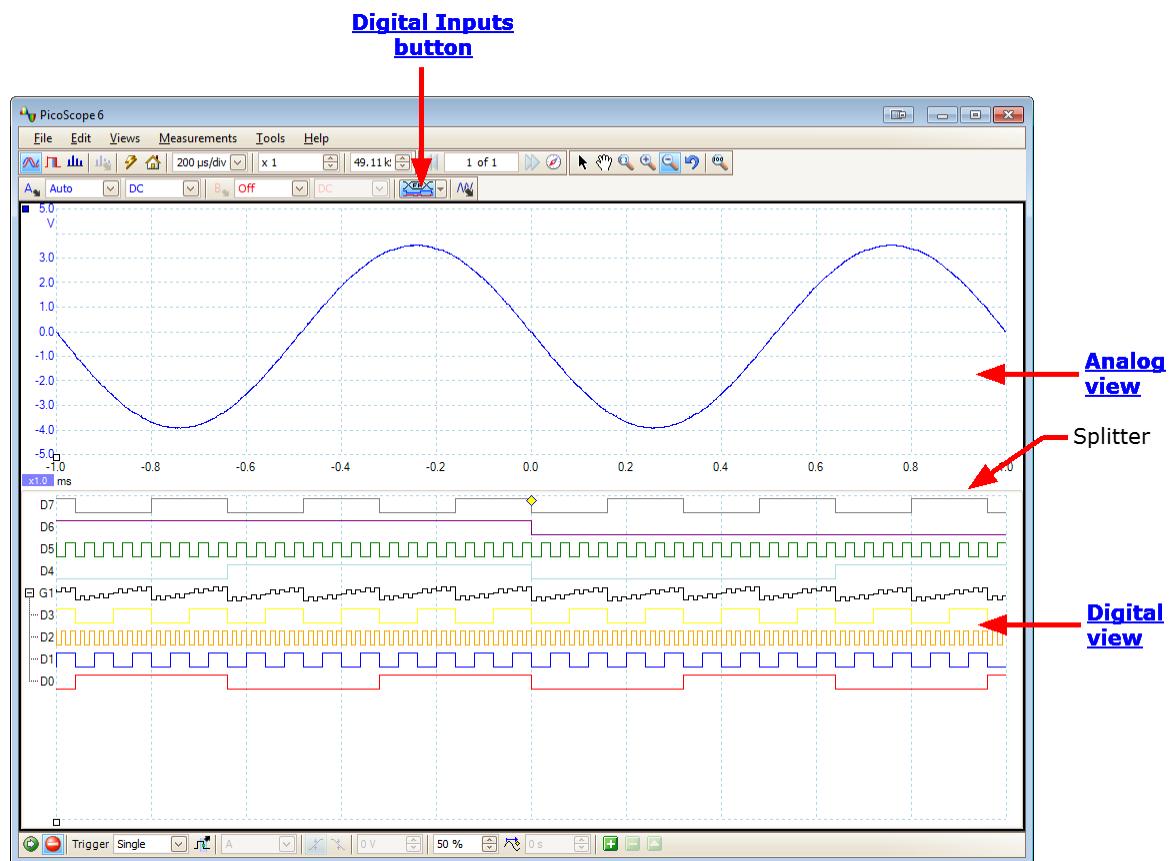
If an overvoltage (a signal outside the measuring range) is detected, the red warning icon (  ) will appear in the upper corner of the PicoScope screen next to the relevant channel's vertical axis.

**Scopes with floating inputs only:** If the BNC shell to chassis voltage exceeds the measurement limit, the Channel LED will turn solid red, and the yellow warning icon (  ) will appear in the upper corner of the PicoScope screen next to the relevant channel's vertical axis. Parts of the waveform will also be missing when the measurement limit is exceeded.

## 5.7 MSO view

Applicability: mixed-signal oscilloscopes ([MSOs](#)) only

The **MSO view** shows mixed analog and digital data on the same timebase.



**Digital Inputs button:** Switches [digital view](#) on and off, and opens the [Digital Setup dialog](#).

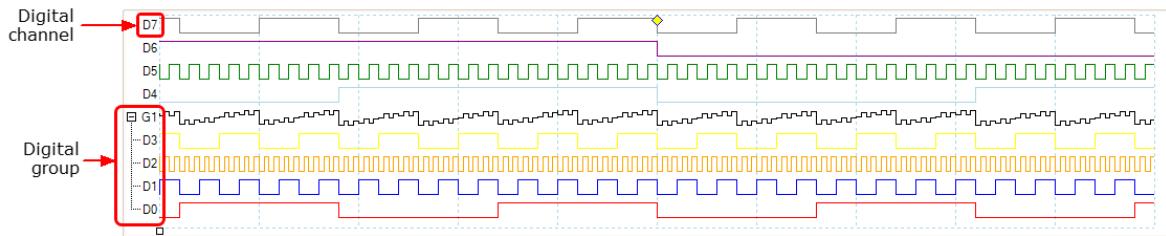
**Analog view:** Shows the analog channels. The same as a standard [scope view](#).

**Digital view:** Shows the digital channels and groups. See [digital view](#).

**Splitter:** Drag up and down to move the partition between analog and digital sections.

### 5.7.1 Digital view

Location: [MSO view](#)



*Note 1: You can right-click on the **digital view** to obtain the [Digital Context Menu](#).*

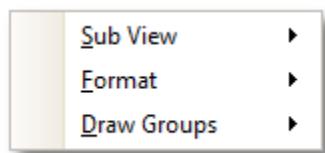
*Note 2: If the **digital view** is not visible when required, check that (a) the [Digital Inputs button](#) is activated and (b) at least one digital channel is selected for display in the [Digital Setup dialog](#).*

**Digital channel:** Displayed in the order in which they appear in the [Digital Setup dialog](#), where they can be renamed.

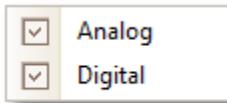
**Digital group:** Groups are created and named in the [Digital Setup dialog](#). You can expand and collapse them in the **digital view** using the  $\square$  and  $\pm$  buttons.

### 5.7.2 Digital context menu

Location: right-click on the [digital view](#)



#### Sub View:

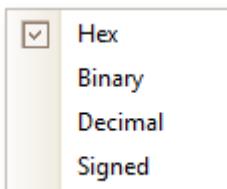


**Analog:** View or hide the [analog scope view](#).

**Digital:** View or hide the [digital scope view](#).

Also available from the [Views menu](#). The numerical format in which group values are displayed in the [digital scope view](#).

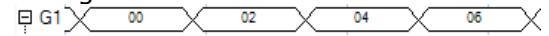
#### Format:



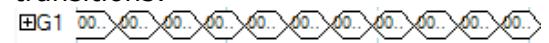
#### Draw Groups:



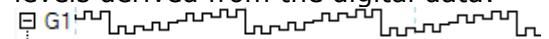
**By Values:** Draw groups with transitions only where the value changes:



**By Time:** Draw groups with transitions spaced equally in time, once per sampling period. You will usually need to zoom in to see the individual transitions:

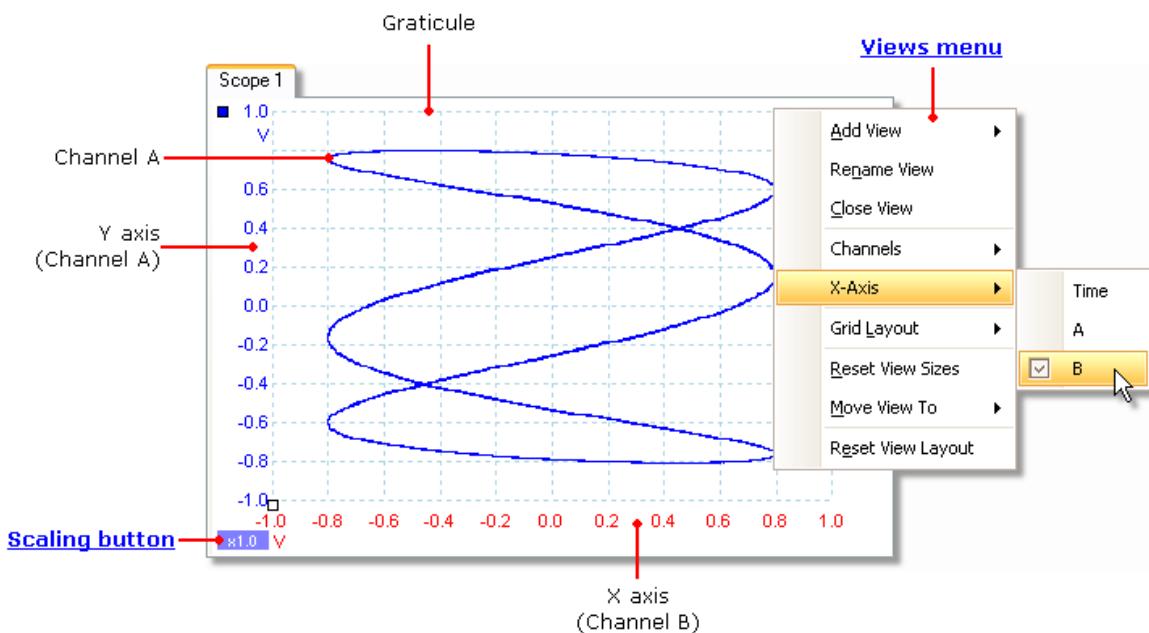


**By Level:** Draw groups as analog levels derived from the digital data:



## 5.8 XY view

An **XY view**, in its simplest form, shows a graph of one channel plotted against another. XY mode is useful for showing relationships between periodic signals (using Lissajous figures) and for plotting I-V (current-voltage) characteristics of electronic components.



In the example above, two different periodic signals have been fed into the two input channels. The smooth curvature of the trace tells us that the inputs are roughly or exactly sine waves. The three loops in the trace show that Channel B has about three times the frequency of Channel A. We can tell that the ratio is not exactly three because the trace is slowly rotating, although you cannot see that in this static picture. Since an XY view has no time axis, it tells us nothing about the absolute frequencies of the signals. To measure frequency, we need to open a [Scope view](#).

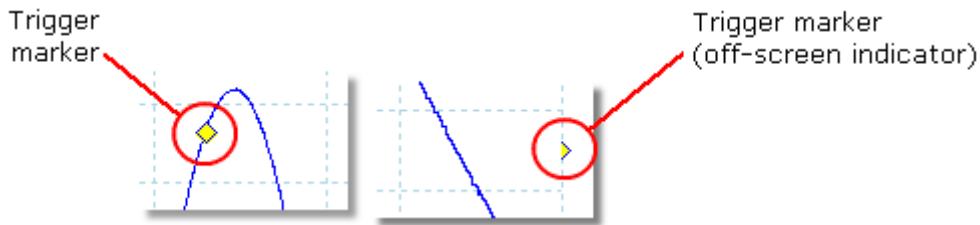
### How to create an XY view

There are two ways to create an XY view.

- Use the **Add View > XY** command on the [Views menu](#). This adds a new XY view to the PicoScope window without altering the original [scope](#) or [spectrum](#) view or views. It automatically chooses the two most suitable channels to place on the X and Y axes. Optionally, you can change the X axis channel assignment using the **X-Axis** command (see below).
- Use the **X-Axis** command on the [Views menu](#). This converts the current scope view into an XY view. It maintains the existing Y axes and allows you to choose any available channel for the X axis. With this method, you can even assign a [math channel](#) or a [reference waveform](#) to the X axis.

## 5.9 Trigger marker

The **trigger marker** shows the level and timing of the trigger point.



The height of the marker on the vertical axis shows the level at which the trigger is set, and its position on the time axis shows the time at which it occurs.

You can move the trigger marker by dragging it with the mouse or, for more accurate control, by using the buttons on the [Triggering toolbar](#).

### Other forms of trigger marker

If the scope view is zoomed and panned so that the trigger point is off the screen, the off-screen trigger marker (shown above) appears at the side of the graticule to indicate the trigger level.

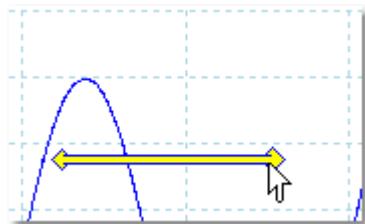
In post-trigger delay mode, the trigger marker is temporarily replaced by the [post-trigger arrow](#) while you adjust the post-trigger delay.

When some [advanced trigger types](#) are in use, the trigger marker changes to a window marker, which shows the upper and lower trigger thresholds.

For more information, see the section on [Trigger timing](#).

## 5.10 Post-trigger arrow

The **post-trigger arrow** is a modified form of the [trigger marker](#) that appears temporarily on a [scope view](#) while you are setting up a post-trigger delay, or dragging the trigger marker after setting up a post-trigger delay. ([What is a post-trigger delay?](#))



The left-hand end of the arrow indicates the trigger point, and is aligned with zero on the time axis. If zero on the time axis is outside the [scope view](#), then the left-hand end of the post-trigger arrow appears like this:

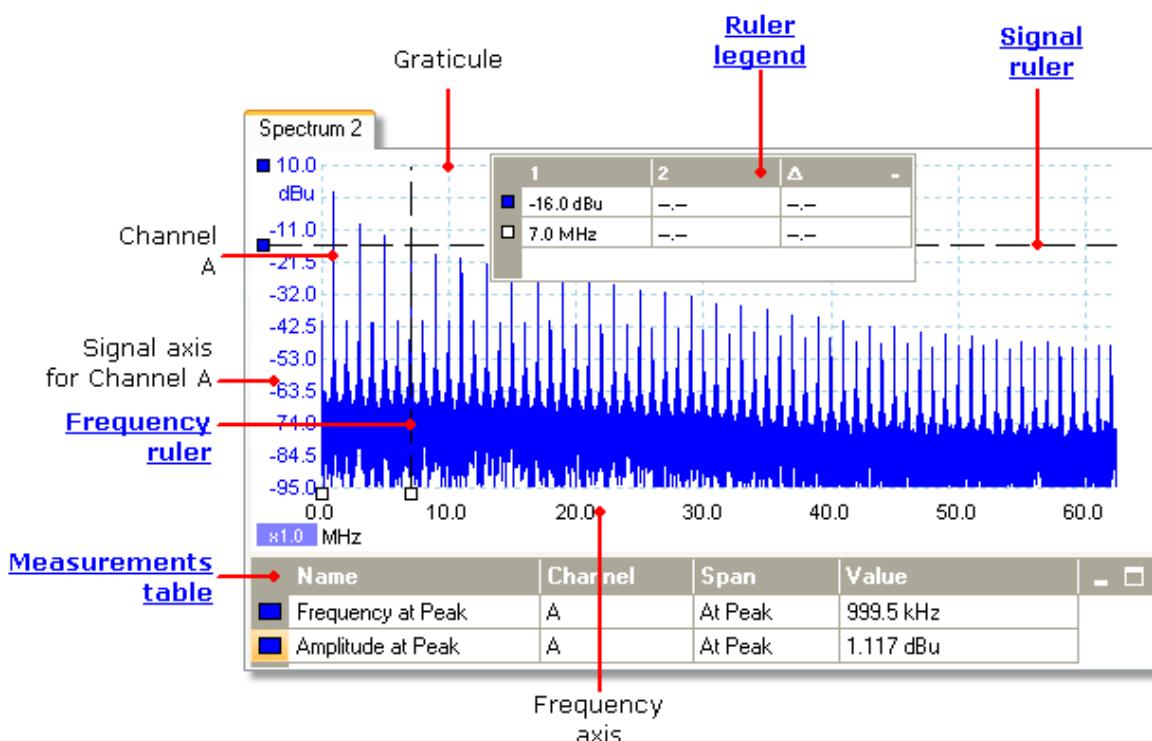


The right-hand end of the arrow (temporarily replacing the [trigger marker](#)) indicates the trigger reference point.

Use the buttons on the [Triggering toolbar](#) to set up a post-trigger delay.

## 5.11 Spectrum view

A **spectrum view** is one view of the data from a scope device. A spectrum is a diagram of signal level on a vertical axis plotted against frequency on the horizontal axis. PicoScope opens with a scope view, but you can add a spectrum view by using the [views menu](#). Similar to the screen of a conventional spectrum analyzer, a spectrum view shows you one or more spectra with a common frequency axis. Each view can have as many spectra as the scope device has channels. Click on one of the labels below to learn more about a feature.



Unlike in the scope view, in the spectrum view the data is not clipped at the limits of the range displayed on the vertical axis, so you can apply axis scaling or offset to see more data. Vertical axis labels are not provided for data outside what is considered to be the 'useful' range, but rulers will still work outside this range.

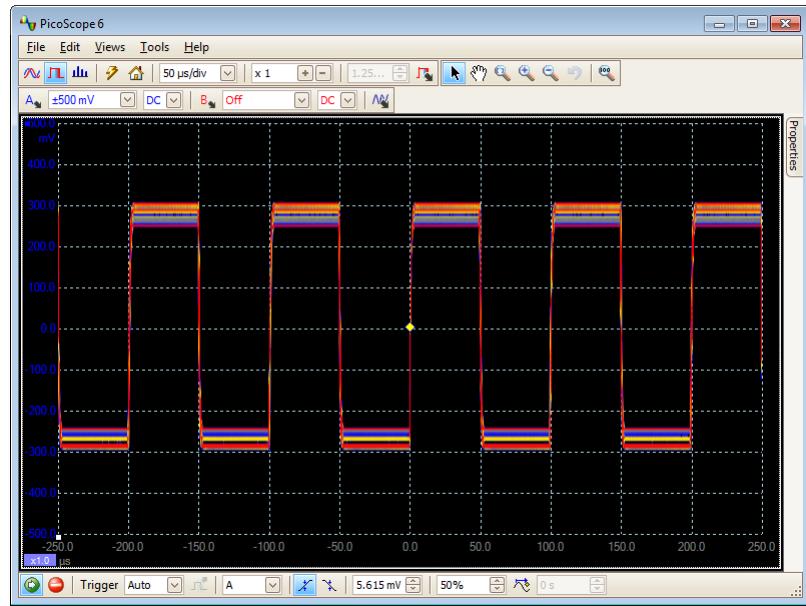
Spectrum views are available regardless of which mode - [Scope Mode](#) or [Spectrum Mode](#) - is active.

For more information, see: [How to set up the spectrum view](#) and [Spectrum Options dialog](#).

## 5.12 Persistence mode

**Persistence mode** superimposes multiple waveforms on the same view, with more frequent data or newer waveforms drawn in brighter colors than older ones. This is useful for spotting glitches, when you need to see a rare fault event hidden in a series of repeated normal events.

Enable persistence mode by clicking the **Persistence Mode button**  on the **Capture Setup toolbar**. With the [persistence options](#) set at their default values, the screen will look something like this:



The colors indicate the frequency of the data. Red is used for the highest-frequency data, with yellow for intermediate frequencies and blue for the least frequent data. In the example above, the waveform spends most of its time in the red region, but noise causes it to wander occasionally into the blue and yellow regions. These are the default colors, but you can change them using the [Persistence Options dialog](#).

This example shows persistence mode in its most basic form. See the [Persistence Options dialog](#) for ways to modify the display to suit your application, and [How to find a glitch using persistence mode](#) for a worked example.

## 5.13 Measurements table

A **measurements table** displays the results of automatic measurements. Each [view](#) can have its own table, and you can add, delete or edit measurements from this table.

Channel	Name	Span	Value	Min	Max	Average	$\sigma$	Capture Count	-
■ A	AC RMS	Whole trace	295.4 mV	295.3 mV	295.6 mV	295.5 mV	105.4 $\mu$ V	20	
■ A	Frequency	Whole trace	10.48 kHz	10.01 kHz	10.48 kHz	10.27 kHz	152 Hz	20	
■ A	Rise Time [80/20%]	Whole trace	1.1 $\mu$ s	1.06 $\mu$ s	1.12 $\mu$ s	1.09 $\mu$ s	15.22 ns	20	

### Measurements table columns

<b>Name</b>	The name of the measurement that you selected in the <a href="#">Add Measurement</a> or <a href="#">Edit Measurement</a> dialog. An "F" after the name indicates that the statistics for this measurement are <a href="#">filtered</a> .
<b>Span</b>	The section of the waveform or spectrum that you want to measure. This is 'Whole trace' by default.
<b>Value</b>	The live value of the measurement, from the latest capture
<b>Min</b>	The minimum value of the measurement since measuring began
<b>Max</b>	The maximum value of the measurement since measuring began
<b>Average</b>	The arithmetic mean of the measurements from the last $n$ captures, where $n$ is set in the <a href="#">General</a> page of the <a href="#">Preferences</a> dialog
<b><math>\sigma</math></b>	The <a href="#">standard deviation</a> of the measurements from the last $n$ captures, where $n$ is set in the <a href="#">General</a> page of the <a href="#">Preferences</a> dialog
<b>Capture Count</b>	The number of captures used to create the statistics above. This starts at 0 when triggering is enabled, and counts up to the number of captures specified in the <a href="#">General</a> page of the <a href="#">Preferences</a> dialog.

### To add, edit or delete measurements

See: [Measurements toolbar](#).

### To change the width of a measurement column

First ensure that the **Column Auto-width** option is not enabled in the [Measurements](#) menu. If necessary, click the option to switch it off. Then drag the vertical separator between column headings to resize the columns, as shown opposite.

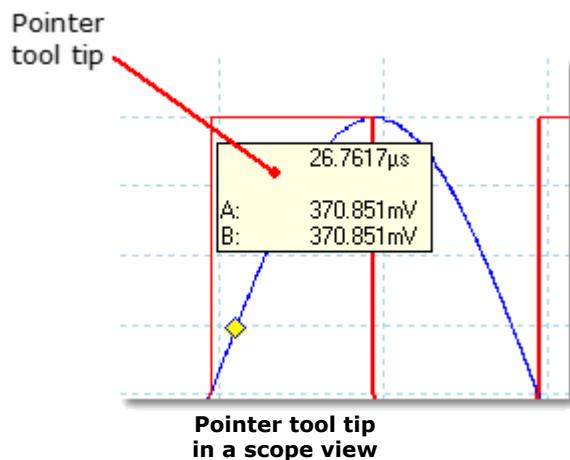
	Max	Average
1	205.8 mV	205.5 mV
2	1 ms	1 ms
3	-8.829 mV	-9.39 mV

### To change the update rate of the statistics

The statistics (**Min**, **Max**, **Average**, **Standard Deviation**) are based on the number of captures shown in the **Capture Count** column. You can change the maximum capture count using the **Capture Size** control in the [General page](#) of the [Preferences](#) dialog.

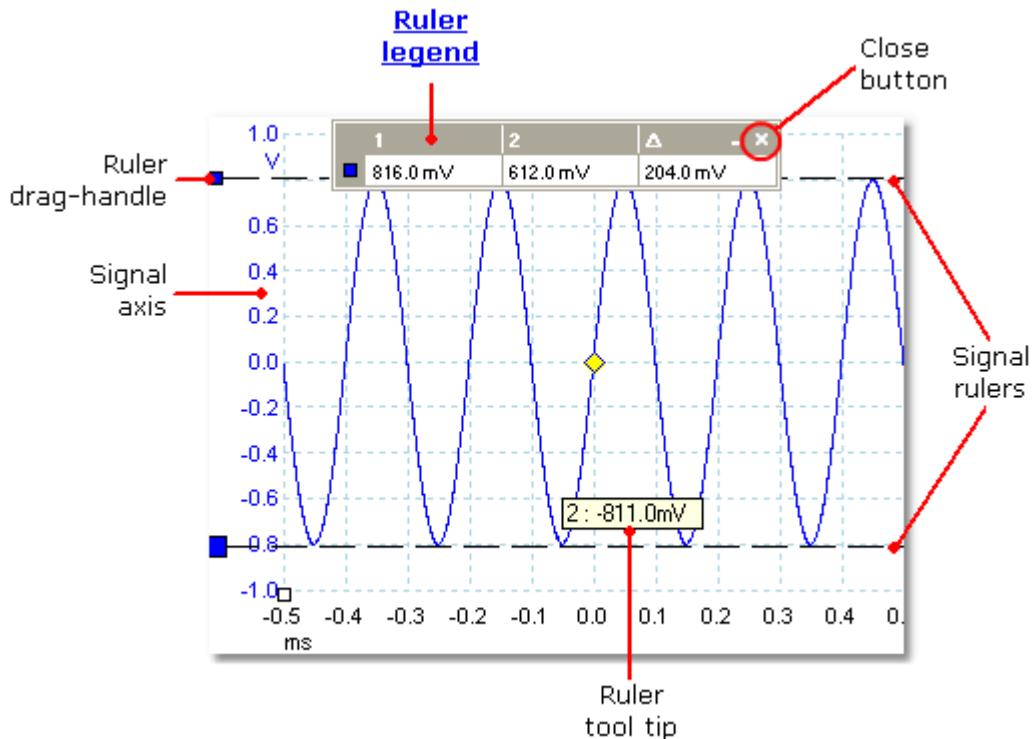
## 5.14 Pointer tool tip

The **pointer tool tip** is a box that displays the horizontal and vertical axis values at the mouse pointer location. It appears temporarily when you click the background of a [view](#).



## 5.15 Signal rulers

The **signal rulers** (sometimes called **cursors**) help you measure absolute and relative signal levels on a [scope](#), [XY](#) or [spectrum view](#).



In the [scope view](#) above, the two colored squares to the left of the vertical axis are the **ruler drag-handles** for channel A. Drag one of these downwards from its resting position in the top left corner, and a **signal ruler** (a horizontal dashed line) will extend from it.

Whenever one or more signal rulers is in use, the [ruler legend](#) appears. This is a table showing all of the signal ruler values. If you close the ruler legend using the **Close** button, all the rulers are deleted.

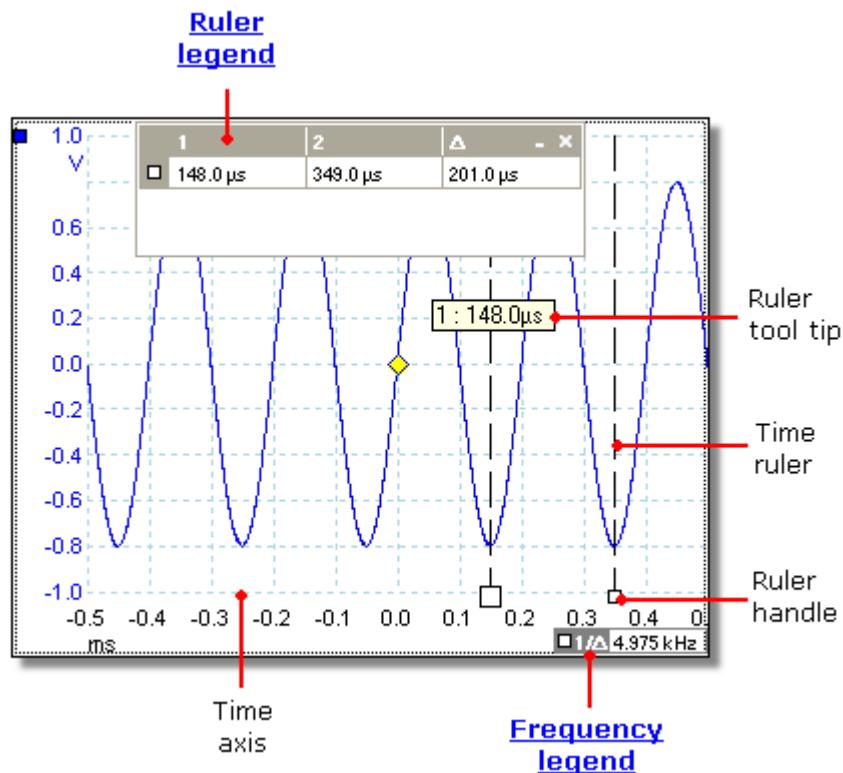
Signal rulers also work in [spectrum](#) and [XY](#) views.

### Ruler tool tip

If you move the mouse pointer over one of the rulers, PicoScope displays a [tool tip](#) with the ruler number and the signal level of the ruler. You can see an example of this in the picture above.

## 5.16 Time rulers

The **time rulers** measure time on a [scope view](#) or frequency on a [spectrum view](#).



In the [scope view](#) above, the two white squares on the time axis are the **time ruler handles**. When you drag these to the right from the bottom left corner, vertical dashed lines called **time rulers** appear. The rulers work in the same way on a [spectrum view](#), but the ruler legend shows their horizontal positions in units of frequency rather than time.

### Ruler tool tip

If you hold the mouse pointer over one of the rulers, as we did in the example above, PicoScope displays a tool tip with the ruler number and the time value of the ruler.

### Ruler legend

The table at the top of the view is the [ruler legend](#). In this example, the table shows that time ruler 1 is at 148.0 microseconds, ruler 2 is at 349.0 microseconds and the difference between them is 201.0 microseconds. Clicking the **Close** button on the ruler legend also deletes all the rulers.

### Frequency legend

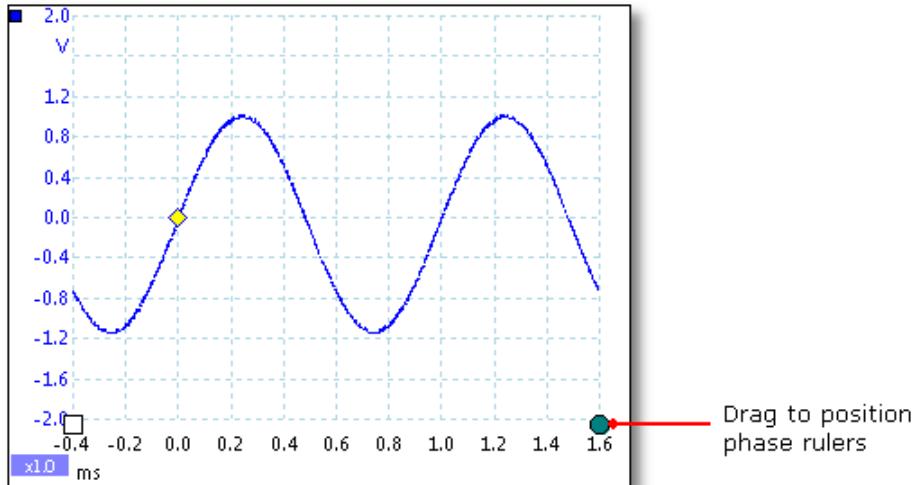
The **frequency legend** in the bottom right-hand corner of a scope view shows  $1/\Delta$ , where  $\Delta$  is the difference between the two time rulers. The accuracy of this calculation depends on the accuracy with which you have positioned the rulers. For greater accuracy with periodic signals, use the [frequency measurement](#) function built in to PicoScope.

## 5.17 Phase (rotation) rulers

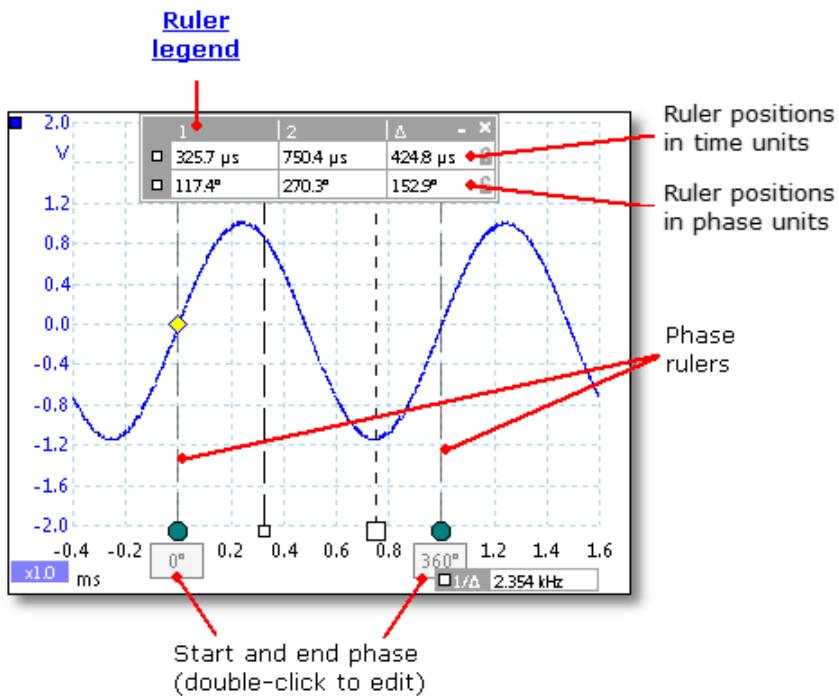
Location: [Scope view](#)

The **phase rulers** (called **rotation rulers** in PicoScope Automotive) help to measure the timing of a cyclic waveform on a [scope view](#). Instead of measuring relative to the trigger point, as [time rulers](#) do, phase rulers measure relative to the start and end of a time interval that you specify. Measurements may be shown in degrees, percent or a custom unit as selected by the [Ruler settings](#) box.

To use the phase rulers, drag the two phase ruler handles onto the waveform from their inactive position as shown below:



When you have dragged both phase rulers into position, the scope view will look like this (we also added two [time rulers](#), for a reason that will be explained later):



In the [scope view](#) above, the two phase rulers have been dragged into place to mark the start and end of a cycle.

The default start and end phase values of 0° and 360° are shown below the rulers and can be edited to any custom value. For example, when measuring timings on a four-stroke engine, it is customary to show the end phase as 720° as one cycle comprises two rotations of the crankshaft.

### Ruler legend

The phase rulers become more powerful when used in conjunction with [time rulers](#). When both types of rulers are used together, as shown above, the [ruler legend](#) displays the positions of the time rulers in phase units as well as time units. If two time rulers are positioned, the legend also shows the phase difference between them. Closing the ruler legend dismisses all rulers including the phase rulers.

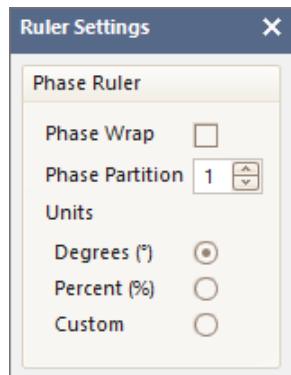
### Ruler options

Options for the **phase (rotation) rulers** are configured by the [Ruler Settings dialog](#), which is called up by the **Rulers button** on the [Advanced Options toolbar](#).

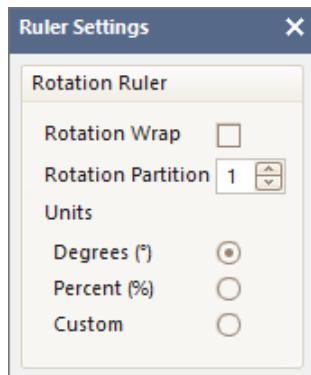
## 5.18 Ruler settings

Location: [Advanced Options toolbar](#) > **Rulers**

The **Ruler Settings** box allows you to control the behavior of the [time rulers](#) and [phase rulers](#) (called **rotation rulers** in PicoScope Automotive).



Ruler Settings in PicoScope



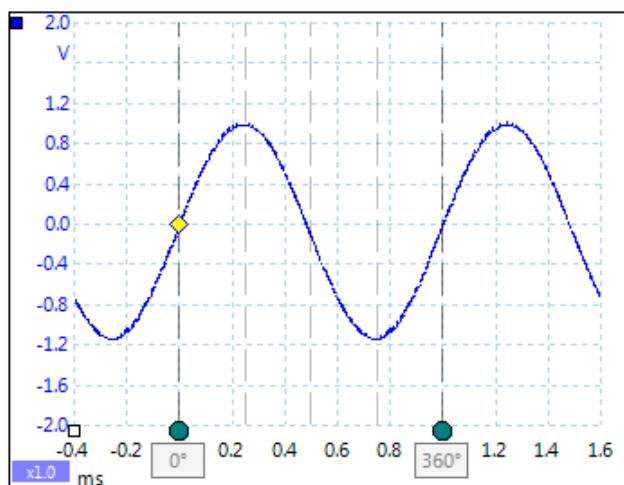
Ruler Settings in PicoScope Automotive

### Phase (Rotation) Wrap

If this box is checked, [time ruler](#) values outside the range set by the [phase \(rotation\) rulers](#) are wrapped back into that range. For example, if the phase (rotation) rulers are set to 0° and 360°, the value of a time ruler just to the right of the 360° phase (rotation) ruler will be 0°, and the value of a time ruler just to the left of the 0° phase (rotation) ruler will be 359°. If this box is unchecked, ruler values are unconstrained.

### Phase (Rotation) Partition

Increasing this value above 1 causes the space between the two phase (rotation) rulers to be partitioned equally into the specified number of intervals. The intervals are marked by broken lines between the phase (rotation) rulers. The lines help you to interpret complex waveforms such as the vacuum pressure of a four-stroke engine with its intake, compression, ignition and exhaust phases, or a commutated AC waveform in a switch mode power supply.



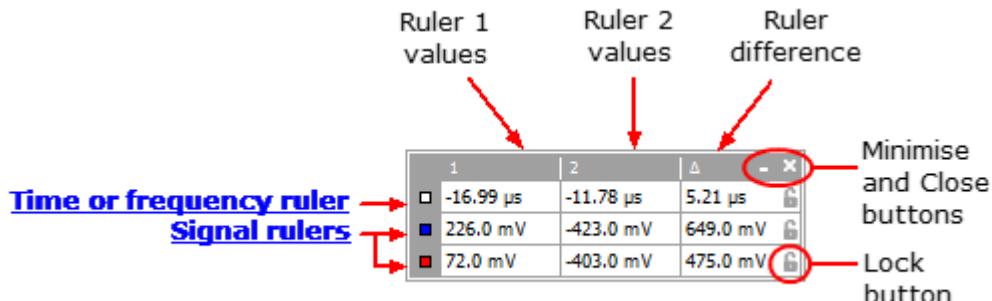
Phase (rotation) rulers with 4 partitions

## Units

You can choose between **Degrees**, **Percent** or **Custom**. **Custom** allows you to enter your own unit symbol or name.

### 5.19 Ruler legend

The **ruler legend** is a box that displays the positions of all the [rulers](#) you have placed on the [view](#). It appears automatically whenever you position a ruler on the view:



## Editing

You can adjust the position of a ruler by editing any value in the first two columns. To insert a Greek  $\mu$  (the *micro* symbol, meaning one millionth or  $\times 10^{-6}$ ), type the letter 'u'.

## Tracking rulers

When two rulers have been positioned on one channel, the **Lock button**  appears next to that ruler in the ruler legend. Clicking this button causes the two rulers to track each other: dragging one causes the other one to follow it, maintaining a fixed separation. The button changes to  when the rulers are locked.

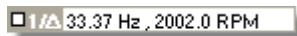
**TIP:** To set up a pair of tracking rulers with a known distance between them, first click the **Lock button**, then edit the two values in the ruler legend so that the rulers are the desired distance apart.

## Phase (rotation) rulers

When [phase rulers](#) (called **rotation rulers** in PicoScope Automotive) are in use, the ruler legend displays additional information.

**See also:** [frequency legend](#).

### 5.20 Frequency legend



The **frequency legend** appears when you have placed two [time rulers](#) on a [scope view](#). It shows  $1/\Delta$  in hertz (the SI unit of frequency, equal to cycles per second), where  $\Delta$  is the time difference between the two rulers. You can use this to estimate the frequency of a periodic waveform, but you will get more accurate results by creating a frequency measurement using the **Add Measurements button** on the [Measurements toolbar](#).

For frequencies up to 1.666 kHz, the frequency legend can also show the frequency in RPM (revolutions per minute). The RPM display can be enabled or disabled in the [\*\*Preferences > Options dialog\*\*](#).

## 5.21 Properties sheet

Location: [Views > View Properties](#)

Purpose: shows a summary of the settings that PicoScope 6 is using

The **Properties sheet** appears on the right-hand side of the PicoScope window.

Properties	
<a href="#">Sampling settings</a>	Sample interval 64 ns Sample rate 15.63 MS/s No. samples 781,250 H/W Resolution 12 bits
<a href="#">Spectrum settings</a>	Window Blackman No. bins 16384 Bin width 476.8 Hz Time gate 2.097 ms
<a href="#">Channel settings</a>	Channel A Range $\pm 10$ mV Coupling DC Res-Enhancement 13.0 Bits Effective Res 11 Bits
<a href="#">Signal generator settings</a>	Signal type Square Frequency 1 kHz Amplitude 1 V Offset 0 V
<a href="#">Time stamp</a>	Capture Date 3/5/13 Capture Time 12:16:37
<a href="#">Capture rate</a>	Capture Rate 14

**No. samples.** The number of samples captured. This may be lower than the number requested in the [Maximum Samples](#) control. A number in brackets is the number of interpolated samples if [interpolation](#) is enabled.

**Window.** The [window function](#) applied to the data before computing the spectrum. This is selected in the [Spectrum options dialog](#).

**Time gate.** The number of samples that PicoScope uses to compute a spectrum is equal to twice the number of bins. This number of samples is expressed as a time interval called the time gate. It is measured from the start of the capture.

**Res-Enhancement (resolution enhancement).** The number of bits, including [resolution enhancement](#), selected in the [Channel Options dialog](#).

**Effective Res (effective resolution; applies to Flexible Resolution oscilloscopes only).** PicoScope tries to use the value specified by the [Hardware Resolution](#) control in the [Capture Setup toolbar](#), but on some voltage ranges the hardware delivers a lower effective resolution. The available resolutions are specified in the data sheet for the scope device.

**Capture rate.** The number of waveforms being captured per second. Shown only in [Persistence Mode](#).

## 5.22 Custom probes

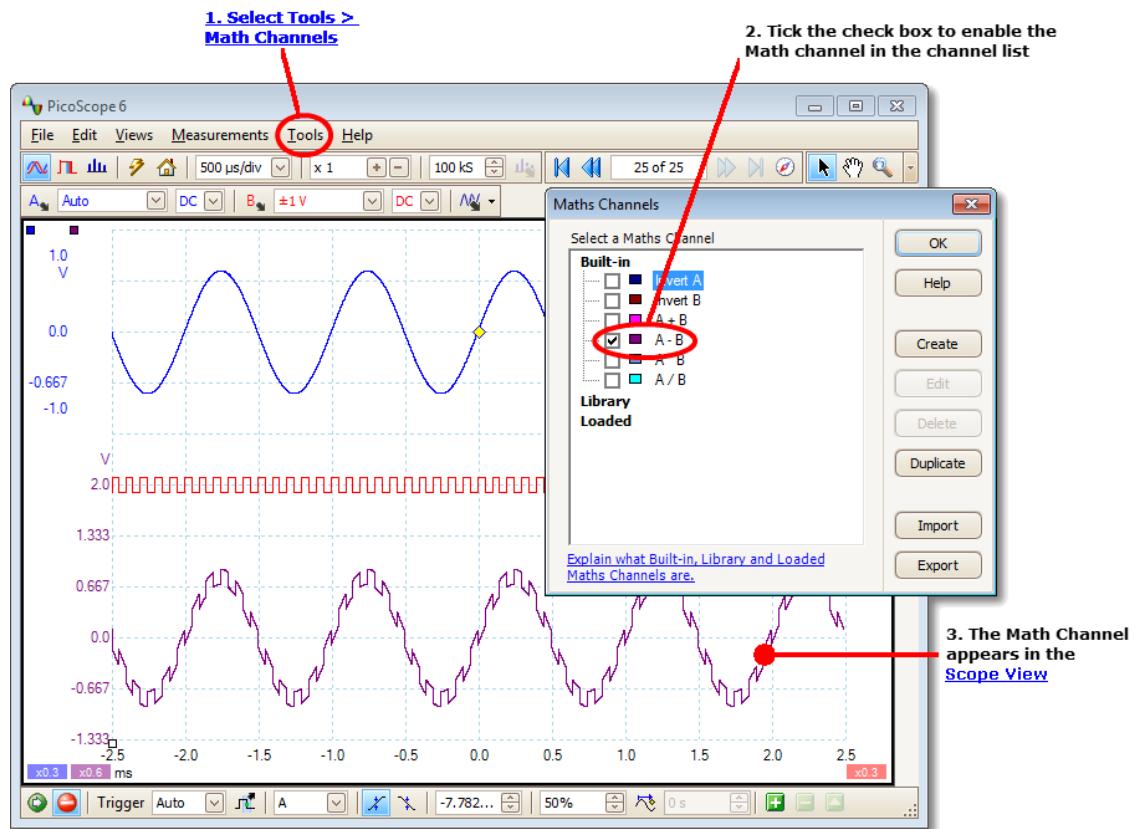
A **probe** is any transducer, measuring device or other accessory that you connect to an input channel of your [scope device](#). PicoScope has a built-in library of common probe types, such as the x1 and x10 voltage probes used with most oscilloscopes, but if your probe is not included in this list you can use the [Custom Probes dialog](#) to define a new one. Custom probes can have any voltage range within the capabilities of the oscilloscope, display in any units, and have either linear or nonlinear characteristics.

Custom probe definitions are particularly useful when you wish to display the probe's output in units other than volts, or to apply linear or nonlinear corrections to the data.

## 5.23 Math channels

A **math channel** is a mathematical function of one or more input signals. It can be displayed in a [scope](#), [XY](#) or [spectrum](#) view in the same way as an input signal, and like an input signal it has its own measurement axis, [scaling and offset button](#) and [color](#). PicoScope 6 has a set of built-in math functions for the most important functions, such as "Invert A", "A+B" and "A-B". You can also define your own functions using the [equation editor](#), or [load predefined math channels from files](#).

Here is a three-step guide to using math channels:



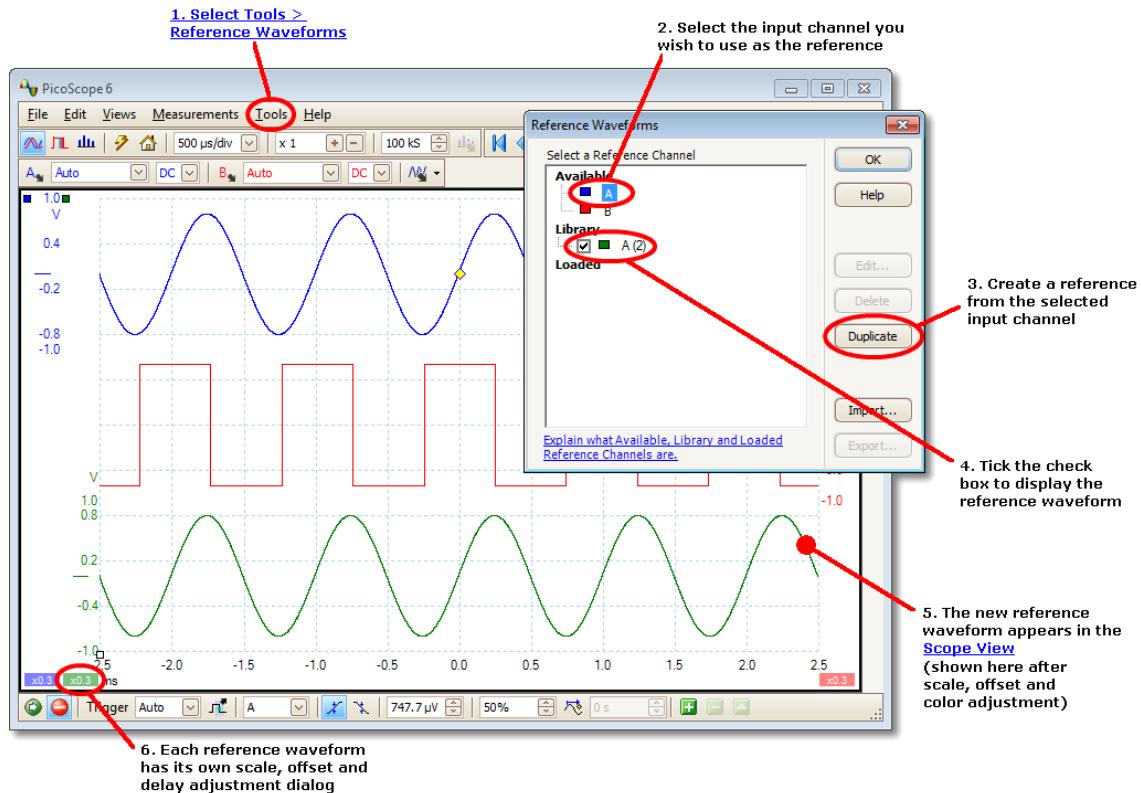
- 1. Tools > Math Channels command.** Click this to open the [Math Channels dialog](#), shown at top right in the picture above.
- 2. Math Channels dialog.** This lists all the available math channels. In the example above, only the built-in functions are listed.
- 3. Math channel.** Once enabled, a math channel appears in the selected [scope](#) or [spectrum](#) view. You can [change its scale and offset](#) as with any other channel. In the example above, the new math channel (bottom) is defined as **A-B**, the difference between input channels A (top) and B (middle).

You may occasionally see a flashing warning symbol - - at the bottom of the math channel axis. This means that the channel cannot be displayed because an input source is missing. For example, this occurs if you enable the **A+B** function while channel B is set to **Off**.

## 5.24 Reference waveforms

A **reference waveform** is a stored version of an input signal. You can create one by right-clicking on the view, selecting the **Reference Waveforms** command and selecting which channel to copy. It can be displayed in a scope or spectrum view in the same way as an input signal, and like an input signal it has its own measurement axis, [scaling and offset button](#) and [color](#). The reference waveform may have fewer samples than the original.

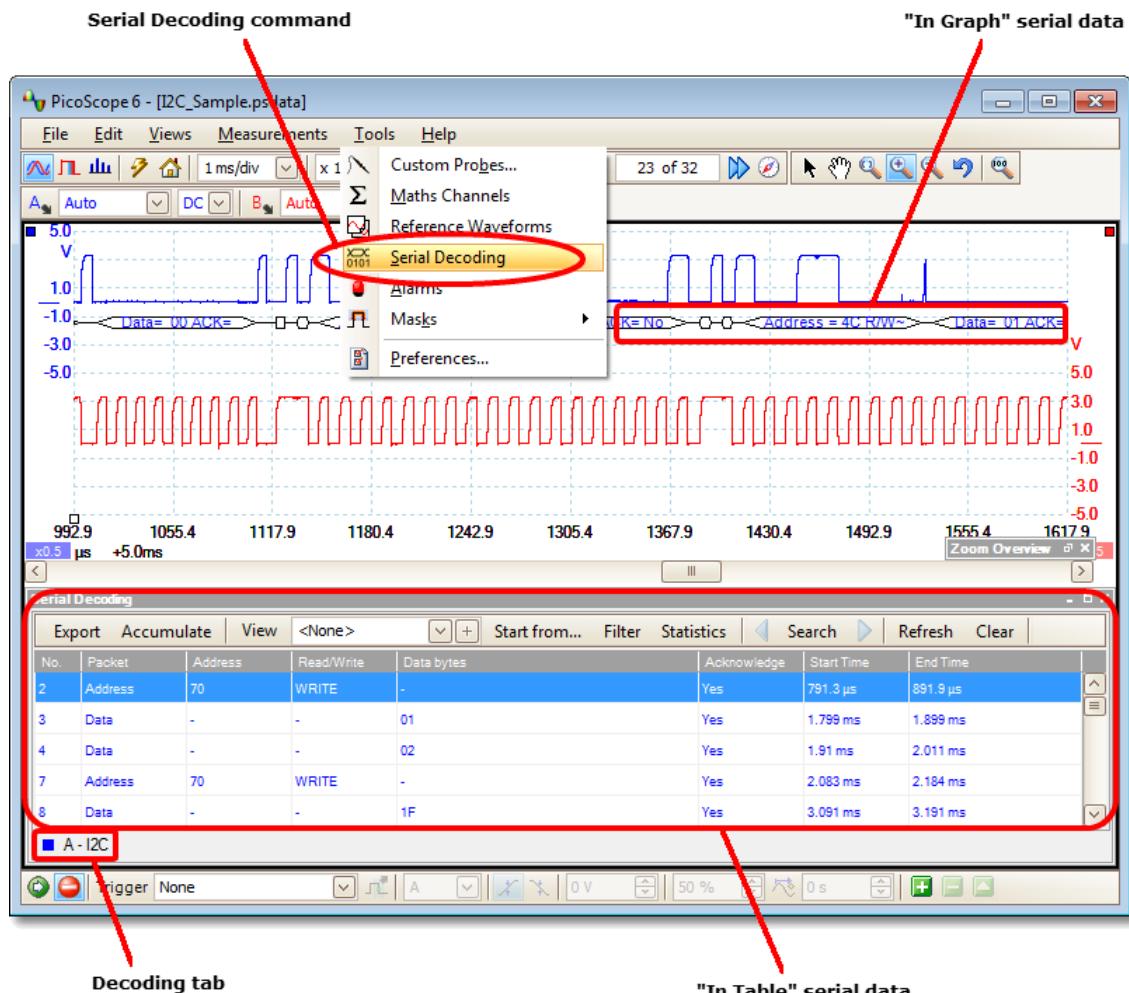
For more control over Reference Waveforms, use the [Reference Waveforms dialog](#) as shown below.



- 1. Reference Waveforms button.** Click this to open the [Reference Waveforms dialog](#), shown on the right in the picture above.
- 2. Reference Waveforms dialog.** This lists all the available input channels and reference waveforms. In the example above, input channels **A** and **B** are switched on, so they appear in the **Available** section. The **Library** section is empty to begin with.
- 3. Duplicate button.** When you select an input channel or reference waveform and click this button, the selected item is copied to the **Library** section.
- 4. Library section.** This shows all your reference waveforms. Each one has a check box that controls whether or not the waveform appears on the display.
- 5. Reference waveform.** Once enabled, a reference waveform appears in the selected [scope](#) or [spectrum](#) view. You can [change its scale and offset](#) as with any other channel. In the example above, the new reference waveform (bottom) is a copy of channel **A**.
- 6. Axis control button.** Opens an [axis scaling dialog](#) allowing you to adjust scale, offset and delay for this waveform.

## 5.25 Serial decoding

You can use PicoScope to decode data from a serial bus such as I<sup>2</sup>C or CAN Bus. Unlike a conventional bus analyzer, PicoScope lets you see the high-resolution electrical waveform at the same time as the data. The data is integrated into the scope view, so there's no need to learn a new screen layout.



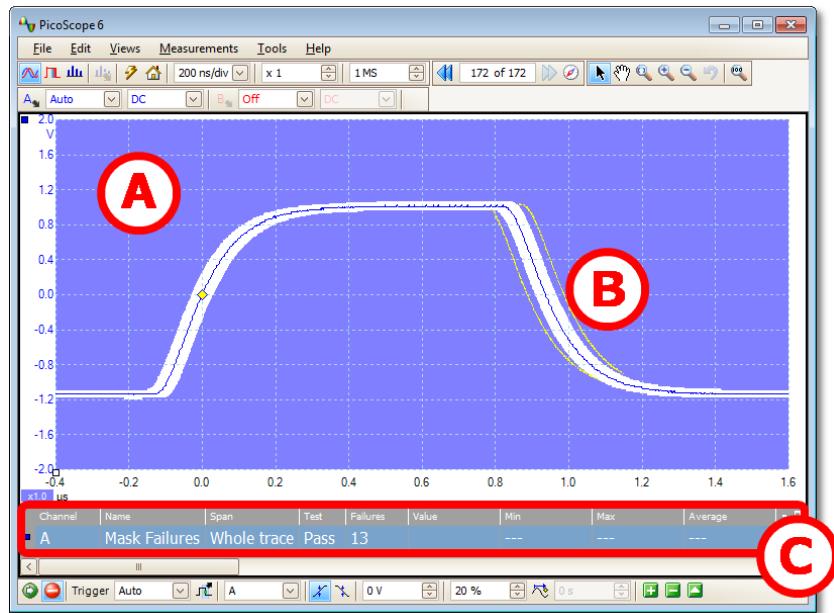
### How to use Serial Decoding

1. Select the [Tools](#) > **Serial Decoding** menu command.
2. Complete the [Serial Decoding dialog](#).
3. Choose to display the data [In Graph](#), [In Table](#), or both.
4. You can decode multiple channels in different formats simultaneously. Use the **Decoding tab** below the "In Table" data table (shown in the picture above) to select which channel of data to display in the table.

## 5.26 Mask limit testing

**Mask limit testing** is a feature that tells you when a waveform or spectrum goes outside a specified area, called a **mask**, drawn on the [scope view](#) or [spectrum view](#). PicoScope can draw the mask automatically by tracing a captured waveform, or you can draw it manually. Mask limit testing is useful for spotting intermittent errors during debugging, and for finding faulty units during production testing.

To begin, go to the main PicoScope menu and select [Tools](#) > [Masks](#) > [Add Masks](#). This opens the [Mask Library dialog](#). When you have selected, loaded or created a mask, the scope view will appear as follows:



### (A) Mask

Shows the allowed area (in white) and the disallowed area (in blue). Right-clicking the mask area and selecting the [Edit Mask](#) command takes you to the [Edit Mask dialog](#). You can change the mask colors with the [Tools](#) > [Preferences](#) > [Colors](#) dialog; add, remove and save masks using the [Masks menu](#); and hide and display masks using the [Views](#) > [Masks menu](#).

### (B) Failed waveforms

If the waveform enters the disallowed area, it is counted as a failure. The part of the waveform that caused the failure is highlighted, and persists on the display until the capture is restarted.

### (C) Measurements table

The number of failures since the start of the current scope run is shown in the [Measurements table](#). You can clear the failure count by stopping and restarting the capture using the [Start/Stop button](#). The measurements table can display [other measurements](#) at the same time as the mask failure count.

## 5.27 Alarms

**Alarms** are actions that PicoScope can be programmed to execute when certain events occur. Use the **Tools > Alarms** command to open the [Alarms dialog](#), which configures this function.

The events that can trigger an alarm are:

- Capture - when the oscilloscope has [captured](#) a complete waveform or block of waveforms.
- Buffers Full - when the [waveform buffer](#) becomes full.
- Mask(s) Fail - when a waveform fails a [mask limit test](#).

The actions that PicoScope can execute are:

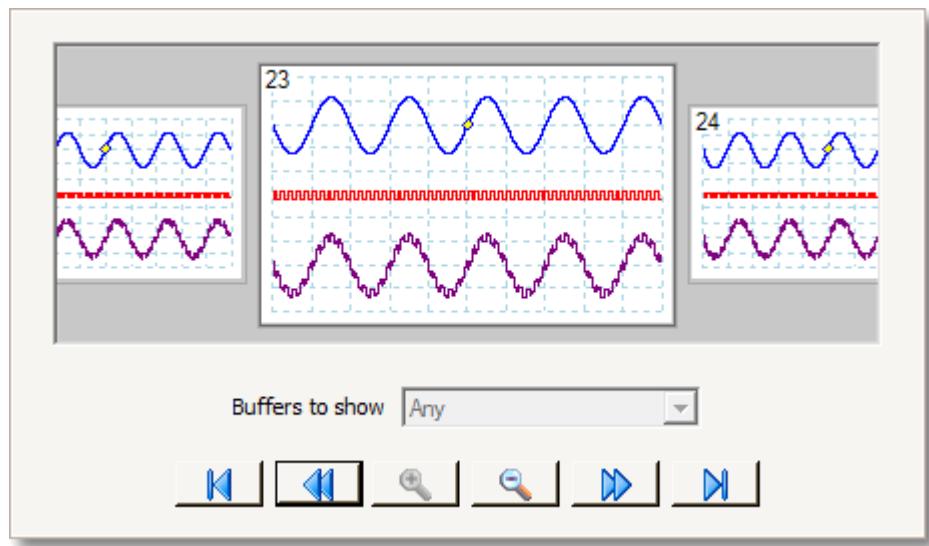
- Beep
- Play Sound
- Stop Capture
- Restart Capture
- Run Executable
- Save Current Buffer
- Save All Buffers

See [Alarms dialog](#) for more details.

## 5.28 Buffer Overview

The PicoScope waveform buffer can hold up to 10,000 waveforms, subject to the amount of available memory in the oscilloscope. The **Buffer Overview** helps you to scroll through the buffer quickly to find the waveform you want.

To begin, click the **Buffer Overview**  button in the [Buffer Navigation toolbar](#). This opens the **Buffer Overview** window:



Click on any one of the visible waveforms to bring it to the front of the overview for closer inspection, or use the controls:

### **Buffers to show**

If any of the channels has a [mask](#) applied, then you can select the channel from this list. The **Buffer Overview** will then show only the waveforms that failed the mask test on that channel.



**Start:**

Scroll to waveform number 1.



**Backward:**

Scroll to the next waveform on the left.



**Zoom in:**

Change the scale of the waveforms in the **Buffer Overview** view. There are three zoom levels:

**Large:** default view. One waveform fills the height of the window.

**Medium:** a medium-sized waveform above a row of small waveforms.

**Small:** a grid of small waveforms. Click on the top or bottom row of images to scroll the grid up or down.



**Zoom out:**

Scroll to the next waveform on the right.



**Forward:**

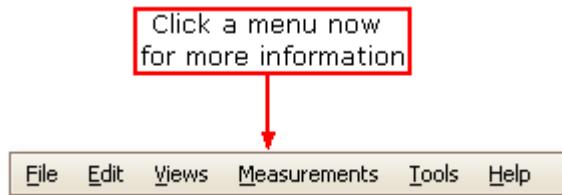
Scroll to the last waveform in the buffer. (The number of

waveforms depends on the [Tools > Preferences > General > Maximum Waveforms](#) setting and on the type of scope connected.)

Click anywhere on the main PicoScope window to close the **Buffer Overview** window.

## 6 Menus

**Menus** are the quickest way to get to PicoScope's main features. The **Menu bar** is always present at the top of the PicoScope main window, just below the window's title bar. You can click any of the menu items, or press the **Alt** key and then navigate to the menu using the arrow keys, or press the **Alt** key followed by the underlined letter in one of the menu items.

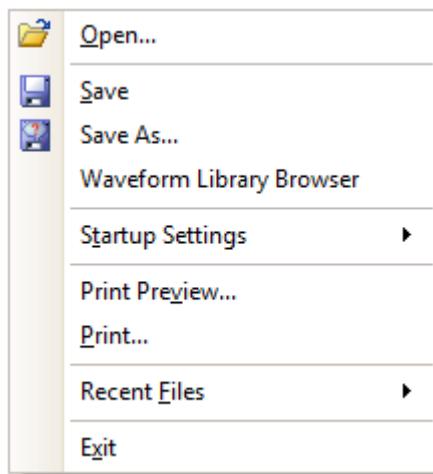


The list of items in the menu bar may vary depending on the windows that you have open in PicoScope.

## 6.1 File menu

Location: [Menu bar](#) > **File**

Purpose: gives access to file input and output operations



**Connect Device.** This command appears only when there is no scope device connected. It opens the [Connect Device dialog](#), which allows you to select the scope device you wish to use.



**Open.** Allows you to select the file you want to open. PicoScope can open `.psdata` and `.psd` files, which contain both waveform data and scope device settings, and `.pssettings` and `.pss` files, which contain only scope device settings. You can create your own files using the **Save** and **Save As...** commands, described below. If the file was saved using a different scope device from the one that is presently connected, PicoScope may need to modify the saved settings to suit the present device.

Hint: Use the **Page Up** and **Page Down** keys to cycle through all the waveform files in the same directory.



**Save.** Saves all waveforms using the filename shown in the title bar. If you haven't entered a filename yet, the [Save As dialog](#) opens to prompt you for one.



**Save As.** Opens the [Save As dialog](#), which allows you to save the settings, waveforms, custom probes and math channels for all views in various formats. Only the waveforms for the mode currently in use ([Scope Mode](#) or [Spectrum Mode](#)) will be saved.

In [persistence mode](#), this command is called **Save Persistence As** and saves only the data for this mode.

**Waveform Library Browser.** Accesses the [Waveform Library Browser](#).

**Startup Settings.** Opens the [Startup Settings menu](#).

**Print Preview.** Opens the **Print Preview window**, which allows you to see how your workspace will be printed when you select the **Print** command.

**Print.** Opens a standard Windows Print dialog, which allows you to choose a printer, set printing options and then print the selected view.

**Recent Files.** A list of recently opened or saved files. This list is compiled automatically, but you can clear it using the **Files** page of the [Preferences](#) dialog.

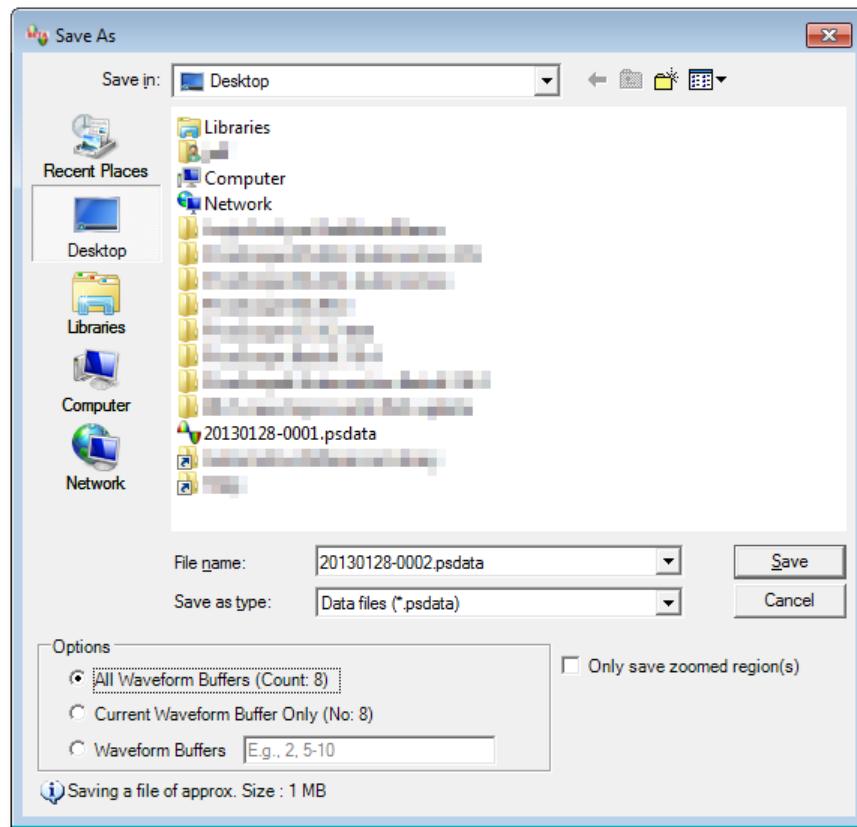
**Exit.** Close PicoScope without saving any data.

### 6.1.1 Save As dialog

Location: [File](#) > **Save All Waveforms As** or **Save Current Waveform As**

Purpose: allows you to save your waveforms and settings (including custom probes and active math channels) to a file in [various formats](#)

**PicoScope Automotive** only: the [Details dialog](#) appears before the **Save As** dialog to allow you to record details of the vehicle and the customer.



Type your chosen file name in the **File name** box, and then select a file format in the **Save as type** box. You can save data in the following formats:

#### **Data files (.psdata)**

Stores waveforms and settings from the current scope device. Can be opened on any computer running PicoScope.

#### **Settings files (.pssettings)**

Stores all settings (but not waveforms) from the current scope device. Can be opened on any computer running PicoScope.

<b>CSV (Comma delimited) files (.csv)</b>	Stores waveforms as a text file with comma-separated values. This format is suitable for importing into spreadsheets such as Microsoft Excel. The first value on each line is the time stamp, and it is followed by one value for each active channel, including currently displayed math channels. <a href="#">(Details)</a>
<b>Text (Tab delimited) files (.txt)</b>	Stores waveforms as a text file with tab-separated values. The values are the same as those in the CSV format. <a href="#">(Details)</a>
<b>Bitmap images (.bmp)</b>	Stores a picture of the waveforms, <a href="#">graticule</a> and <a href="#">rulers</a> in Windows BMP format. The image is 800 pixels wide by 600 pixels high, in 16 million colors, and uncompressed. BMP files are suitable for importing into Windows desktop-publishing programs.
<b>GIF images (.gif)</b>	Stores the waveforms, <a href="#">graticule</a> and <a href="#">rulers</a> in Compuserve GIF format. The image is 800 pixels wide by 600 pixels high, in 256 colors, and compressed. GIF files are widely used to illustrate web pages.
<b>Animated GIF image (.gif)</b>	Creates an animated GIF that displays all of the waveforms in the buffer in sequence. Each waveform is formatted as in the single GIF format described above.
<b>PNG images (.png)</b>	Stores the <a href="#">graticule</a> , <a href="#">rulers</a> and waveforms in PNG format. The image is 800 pixels wide by 600 pixels high, in 16 million colors, with lossless compression.
<b>MATLAB 4 files (.mat)</b>	Stores the waveform data in <a href="#">MATLAB 4 format</a> .
<b>JPEG (.jpg)</b>	Stores the <a href="#">graticule</a> , <a href="#">rulers</a> and waveforms in JPG format. The image is 800 pixels wide by 600 pixels high, in 16 million colors, with lossy compression.

## Options

The first three options control what happens when the [waveform buffer](#) contains more than one waveform:

### All waveform buffers

Save all of the waveforms in the selected file format. If the file format is PSDATA, all of the waveforms are collected in a single file. You can then load this into PicoScope and step through the waveforms using the [buffer navigation controls](#). If the selected file format does not support multiple waveforms, PicoScope creates a new directory containing multiple files.

**Current waveform buffer only**

Save the single waveform that is currently on view.

**Waveform buffers**

Save the specified list or range of waveforms. Each waveform is identified by its index number. For example:

1, 2, 9, 10  
2, 5-10

**Only zoomed regions**

If the waveform is zoomed horizontally, save only the visible portion.

## 6.1.1.1 File formats for exported data

PicoScope 6 can export raw data in either text or binary format: -

**Text-based file formats**

- Easy to read without special tools
- Can be imported into standard spreadsheet applications
- Files are very large if there are many samples in the data (so files are limited to about 1 million values per channel)

[Text file format details](#)**Binary file format**

- Files remain relatively small and can even be compressed in some situations (this means that the amount of saved data is unlimited)
- Either a special application is required to read the files or the user must write a program to read the data from the file

If you need to save more than 64 K values per channel, then you must use a binary file format such as the MATLAB® MAT-file format.

[Binary file format details](#)**Data types for storing PicoScope 6 data**

Regardless of whether the data types were loaded from a binary file or from a text-based file, we recommend the following data formats for storing the values loaded from a PicoScope 6 data file: -

- Sampled data (such as voltages) should use 32-bit single-precision floating-point data types.
- Times should use 64-bit double-precision floating-point data types.

## 6.1.1.1.1 Text formats

Text-format [files exported by PicoScope 6](#) are encoded in [UTF-8](#) format by default. This is a popular format which is capable of representing a huge range of characters, whilst still retaining some compatibility with the ASCII character set if only standard Western European characters and numbers are used in the file.

**CSV (comma-separated values)**

CSV files store data in the following format: -

```
Time, Channel A, Channel B
(μs), (V), (V)
-500.004, 5.511, 1.215
-500.002, 4.724, 2.130
-500, 5.552, 2.212
...
```

There is a comma after each value on a line to represent a column of data and a carriage return at the end of the line to represent a new row of data. The 1 million values per channel limit prevents excessively large files being created.

**Note.** CSV files are not the best choice of format if you are working in a language that uses the comma character as the decimal point. Instead, try using the tab-delimited format which works in almost the same way.

### Tab-delimited

Tab-delimited files store data in the following format: -

Time ( $\mu$ s)	Channel A (V)	Channel B (V)
500.004	5.511	1.215
-500.002	4.724	2.130
-500	5.552	2.212
...		

The files have a tab character after each value on a line to represent a column of data and a carriage return at the end of the line to represent a new row of data. These files work in any language and are a good choice for sharing data internationally. The 1 million values per channel limit prevents excessively large files being created.

#### 6.1.1.1.2 Binary formats

PicoScope 6 can [export data](#) in **version 4 of the .mat binary file format**. This is an open format and the full specification is freely available from the [www.mathworks.com](http://www.mathworks.com) website. PicoScope 6 saves data into the MAT-File format in a specific way, which is detailed below.

### Importing into MATLAB

Load the file into your workspace using this syntax: -

```
load myfile
```

Each channel's data is stored in an array variable named by the channel. So, the sampled data for channels A to D would be in four arrays named **A**, **B**, **C** and **D**.

There is only one set of time data for all channels and this is loaded in one of two possible formats:

1. A start time, an interval and a length. The variables are named **Tstart**, **Tinterval** and **Length**.
2. An array of times (sometimes used for ETS data). The time array is named **T**.

If the times are loaded in as **Tstart**, **Tinterval** and **Length** then you can use the following command to create the equivalent array of times: -

```
T = [Tstart : Tinterval : Tstart + (Length - 1) * Tinterval];
```

*Note:* The size of the largest file that MATLAB can open depends on the resources of the computer. It is therefore possible for PicoScope to create a MATLAB file that some installations of MATLAB might be unable to open. Please be aware of this risk when saving critical data.

### Exploring the file format

The full file specification, available from [www.mathworks.com](http://www.mathworks.com), is comprehensive so this guide does not describe the entire format. Instead, this guide describes enough of the format to allow you to get data from the file and use it in your own program.

The variables described above (under [Importing into Matlab](#)) are stored in a series of data blocks, each preceded by a header. Each variable has its own header and data block and the corresponding variable names are stored with them (such as **A**, **B**, **Tstart**). The following sections describe how to read each variable from the file.

The order of the data blocks is not specified, so programs should look at the variable names to decide which variable is currently being loaded.

### ● The header

The file consists of a number of data blocks preceded by 20-byte headers. Each header contains five 32-bit integers (as described in the table below).

Bytes	Value
0 – 3	Data format (0, 10 or 20)
4 – 7	Number of values
8 – 11	1
12 – 15	0
16 – 19	Name length

### ● Data format

The 'Data format' in the first 4-bytes describes the type of numerical data in the array.

Value	Description
0	Double (64-bit floating point)
10	Single (32-bit floating point)
20	Integer (32-bit)

### ● Number of values

The 'Number of values' is a 32-bit integer describing the number of numerical values in the array. This value may be 1 for variables that only describe one value; but for arrays of samples or times, expect this to be a large number.

### ● Name length

The 'Name length' is the length of the name of the variable as a null-terminated 1-byte per character ASCII string. The last null terminating character ('\0') is included in the 'Name length' so if the variable name is "TStart" (same as 'TStart\0') then the name length will be 7.

### ● The data block

The data block begins with the name of the variable (such as **A**, **Tinterval**) and you should read in the number of bytes described by the 'Name length' part of the header (not forgetting that the last byte in the string is '\0' if your programming language needs to take account of this).

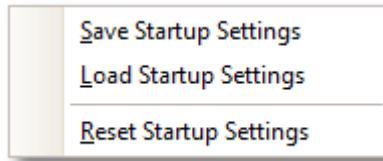
The remaining part of the data block is the actual data itself, so read in the number of values described in the 'Number of values' part of the header. Remember to take account of the size of each value as described in the 'Data format' part of the header.

Channel data such as voltages, in variables such as **A** and **B**, are stored as 32-bit single-precision floating-point data types. Times such as **Tstart**, **Tinterval** and **T** are stored as 64-bit double-precision floating-point data types. **Length** is stored as a 32-bit integer.

### 6.1.2 Startup Settings menu

Location: **File** > **Startup Settings**

Purpose: allows you to load, save and restore the PicoScope 6 startup settings



**Save Startup Settings.** Saves your current settings ready for when you next select

**Load Startup Settings.** These settings are remembered from one session of PicoScope 6 to the next.

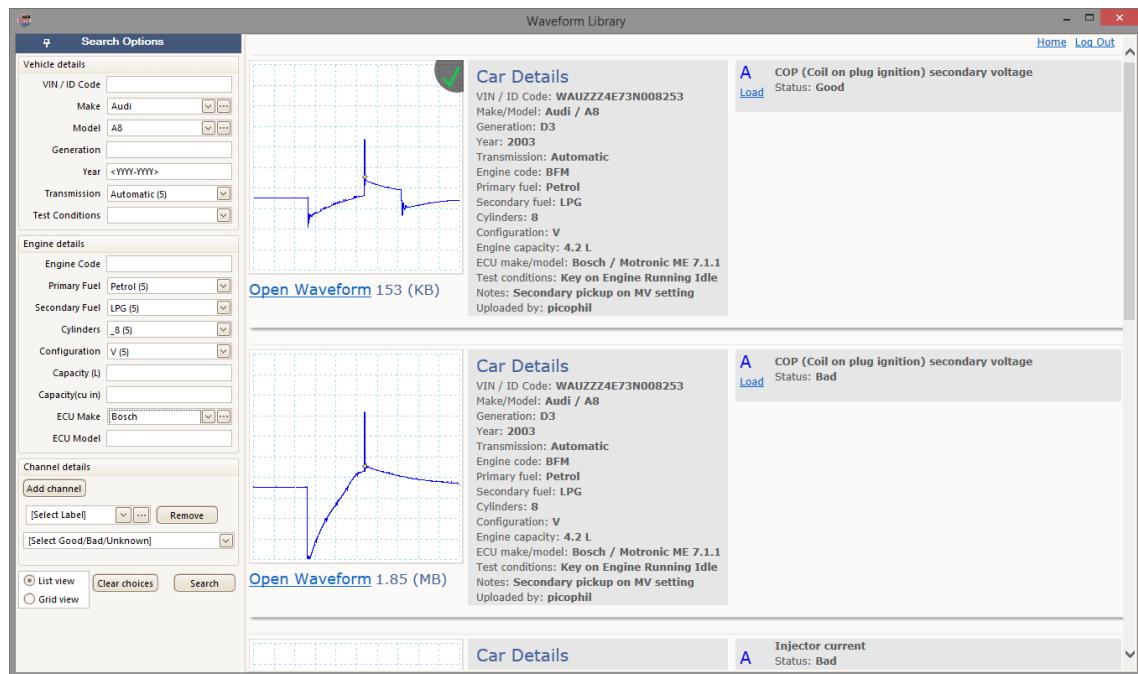
**Load Startup Settings.** Returns to the settings you created with the **Save Startup Settings** command.

**Reset Startup Settings.** Deletes the startup settings you created with the **Save Startup Settings** command, and restores the installation default settings.

### 6.1.3 Waveform Library Browser

Location: **File > Waveform Library Browser (PicoScope Automotive only)**

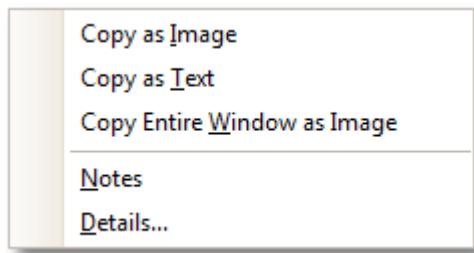
Purpose: allows you to search hundreds of user-uploaded waveforms by entering various fields of required data. Once found, the waveform can be previewed, opened, or even its individual channels used as reference waveforms on your PicoScope screen.



## 6.2 Edit menu

Location: [Menu bar](#) > **Edit**

Purpose: gives access to the clipboard-related and note editing functions



**Copy as Image.** Copies the active view to the clipboard as a bitmap. You can then paste the image into any application that accepts bitmap images.

**Copy as Text.** Copies the data in the active view to the clipboard as text. You can paste the data into a spreadsheet or other application. The text format is the same as that used by the [Save As dialog](#) when you select the `.txt` format.

**Copy Entire Window as Image.** This copies a picture of the PicoScope window to the clipboard. It is provided as an alternative equivalent to pressing Alt-PrtScn for users of laptops without a PrtScn key. You can paste the picture into any application that can display pictures, such as a word processor or desktop publishing program.

**Notes.** Opens a [Notes area](#) at the bottom of the PicoScope window. You can type or paste your own notes in this area.

**Details.** [PicoScope Automotive only] Opens the [Details dialog](#), which allows you to enter or view vehicle details, customer details, [Notes](#) and [Channel Labels](#).

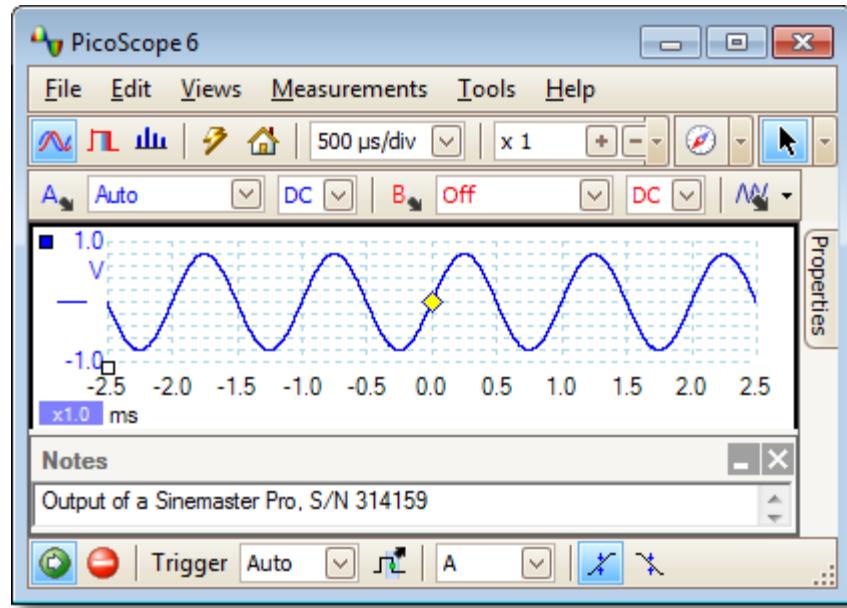
### 6.2.1 Notes

Location: [Edit](#) > **Notes**

[Edit](#) > [Details](#) (PicoScope Automotive only)

[Advanced Options toolbar](#) > **Notes** button

Purpose: provides a text box for typing your own notes



A **Notes** area can be displayed at the bottom of the PicoScope window. You can enter any text you wish in this area. You can also copy text from another program and paste it here. This text is included when you save the waveform as a file.

### 6.2.2 Channel Labels (PicoScope Automotive only)

Location: [\*\*Edit\*\*](#) > [\*\*Details\*\*](#)  
[\*\*Advanced Options toolbar\*\*](#) > [\*\*Channel Labels button\*\*](#)

Purpose: allows you to supply and view information on the waveforms

Channel Labels			
A	Injector current	looks like stuck injector	Bad
B	Injector voltage	normal	Good
C	Ignition coil primary volta	ask Phil	Unknown
D	Ignition coil secondary vc	HT too low	Bad

**Channel Labels** can be displayed at the bottom of the PicoScope window.

**Label:** Select a standard label from the drop-down list or type any text in the box.

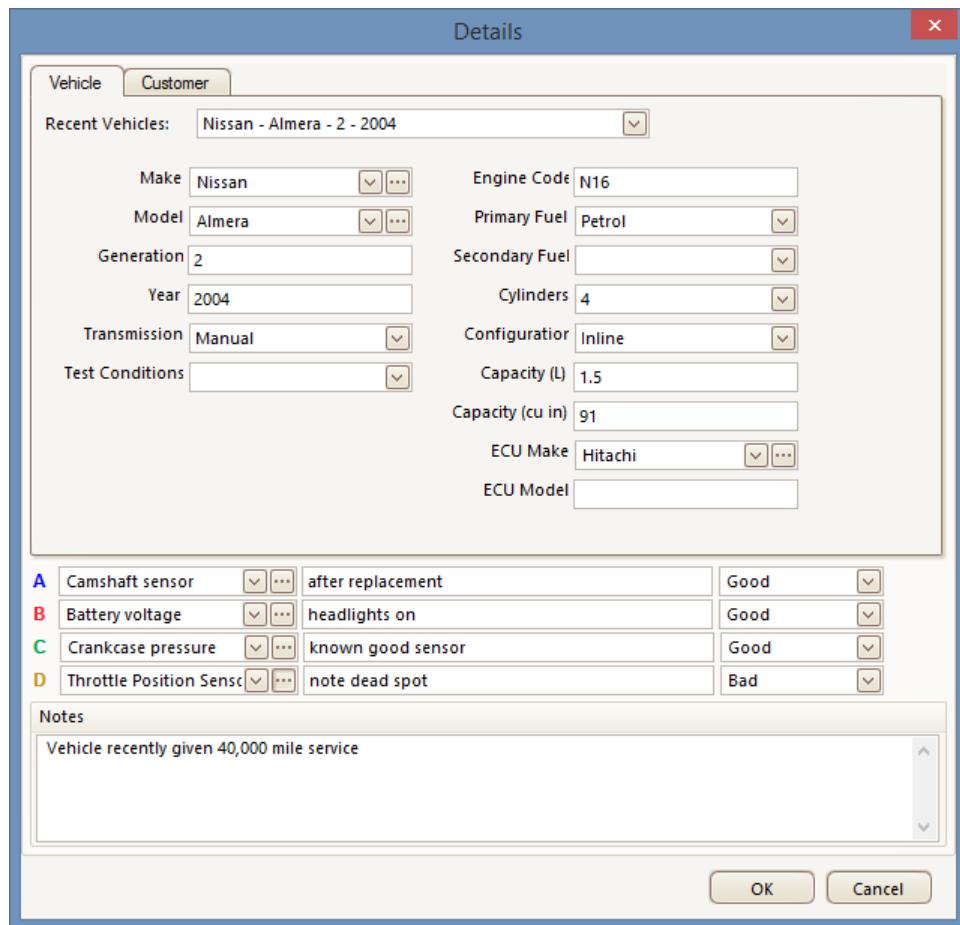
**Description:** Type any text in the box to describe the waveform.

**Status:** Select **Good**, **Bad** or **Unknown**. This helps other engineers to identify whether the waveform is from a working engine or a faulty one.

### 6.2.3 Details dialog (PicoScope Automotive only)

Location: **File > Save As**  
**Edit > Details**

Purpose: allows you to record channel labels, and vehicle and customer details, before saving a file.

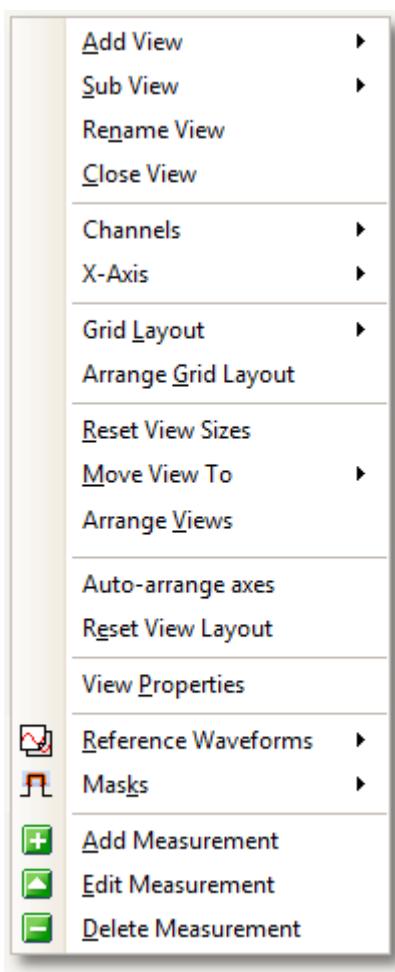


**Details dialog showing vehicle information. Customer information (not shown) can also be entered.**

All of this information will be saved in the PicoScope data file. It can then be viewed in PicoScope using the **Edit > Details** menu command.

Additionally, the **Notes** and **Channel Labels** can be displayed at the bottom of the PicoScope window by clicking the **Notes** and **Channel Labels** buttons in the lower toolbar.

## 6.3 Views menu



**Location:** [Menu bar](#) > **Views**, or right-click on a [view](#)

**Purpose:** controls the layout of the current [view](#), which is a rectangular area of the PicoScope window that display scope, spectrum or other kinds of data

The contents of the **Views menu** may vary depending on where you click and how many views are open. If the current view contains a [Measurements table](#), a combined [Measurements menu](#) and **Views menu** will appear.

**Add View:** Add a view of the selected type ([scope](#), [XY](#) or [spectrum](#)). In automatic grid layout mode (the default), PicoScope rearranges the grid to make room for the new view, up to a limit of four views. Any further views will be added as tabs in existing [viewports](#). If you have selected a fixed grid layout, PicoScope will not change it.

**Sub View:** ([Mixed signal oscilloscopes](#) only) Switch the [analog view](#) and [digital view](#) on and off independently.

**Rename View:** Change the standard 'Scope' or 'Spectrum' label to a title of your choice.

**Close View:** Remove a view from the PicoScope window. In automatic grid layout mode (the default), PicoScope rearranges the grid to make the best use of the remaining space. In fixed grid layout mode (if you have selected a grid fixed layout), PicoScope will not change the grid.

**Channels:** Select which channels are visible in the current view. Each view, when created, shows all the input channels, but you can switch them on and off using this command. Only the input channels that are enabled (not set to "Off" in the [Channel Setup Toolbar](#)) are available for viewing. The **Channels** menu also list [math channels](#) and [reference waveforms](#). You can select up to 8 channels in any view.

**X-Axis:** Select any suitable channel to drive the X axis. By default, the X axis represents time. If you select an input channel instead, the scope view will become an [XY view](#) that plots one input against another. A quicker way to create an XY view is to use the **Add View** command (see above).

**Grid Layout:** The grid layout defaults to "Automatic" mode, in which PicoScope automatically arranges views in a grid. You can also select one of the standard grid layouts or create a custom layout, which PicoScope will preserve as you add or remove views.

**Arrange Grid Layout:** Adjust the grid layout to fit the number of views. Moves any tabbed views to empty viewports. Overrides any previous choice of grid layout.

**Reset View Sizes:** If you have resized any of the views by dragging the vertical or horizontal separator bars between viewports, this option resets all the viewports to their original sizes.

**Move View To:** Move a view to a specified viewport. You can achieve the same effect by dragging the view by its name tab and dropping it in a new viewport. See [How to move a view](#).

**Arrange views:** If multiple views are stacked in the same viewport, move them into their own viewports.

**Auto-arrange axes:** Scale and offsets all traces to fill the view and avoid overlaps.

**Reset View Layout:** Reset the scale factor and offset of the selected view to their default values.

**View Properties:** Show the [Properties sheet](#), which lists scope settings that are normally hidden.

**Reference Waveforms:** Copy one of the available channels to a new [Reference Waveform](#) and add it to the view.

**Masks:** Select which masks (see [Mask Limit Testing](#)) are visible.

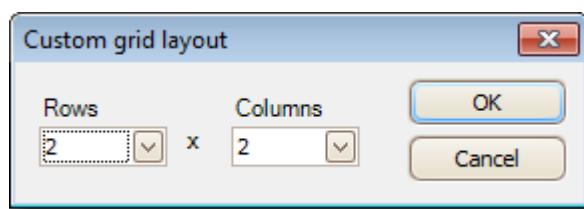
**Add Measurement:** See [Measurements menu](#).

**Edit Measurement:**

**Delete Measurement:**

### 6.3.1 Custom grid layout dialog

Location: right-click on view > [Views menu](#) > **Grid Layout** > **Custom layout...**  
or [Views](#) > **Grid Layout**  
Purpose: if the **Grid Layout** section of the [Views menu](#) does not contain the layout you want, this dialog gives further options

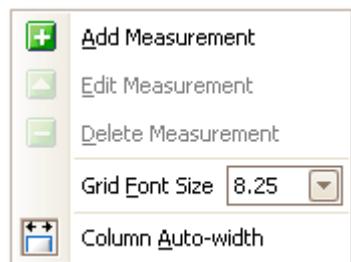


You can lay out the [view](#) grid with any number of rows and columns up to 4 by 4. You can then drag the views to different locations in the grid.

## 6.4 Measurements menu

Location: [Menu bar](#) > **Measurements**

Purpose: controls the [Measurements table](#)



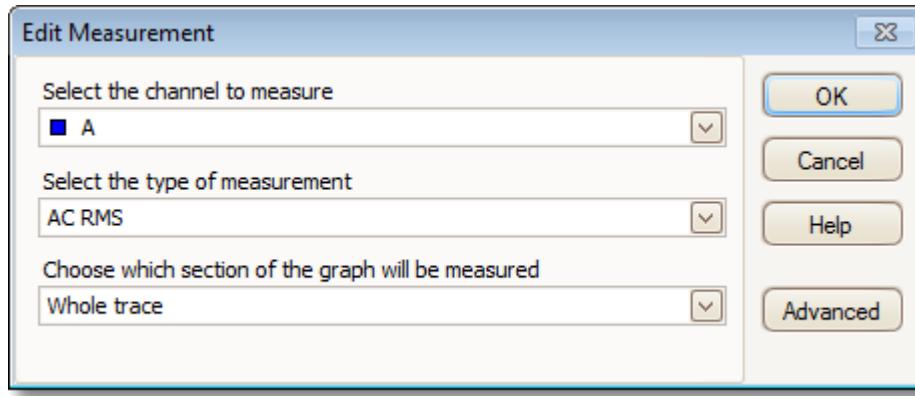
-  **Add measurement.** Adds a row to the [measurements table](#), and opens the [Edit Measurement Dialog](#). You can also find this button on the [Measurements toolbar](#).
-  **Edit measurement.** This takes you to the [Edit Measurement Dialog](#). You can find this button on the [Measurements toolbar](#), or you can edit a measurement by double-clicking on a row of the [measurements Table](#).
-  **Delete measurement.** Removes the selected row from the [measurements table](#). You can also find this button on the [Measurements toolbar](#).
-  **Grid font size.** Sets the font size for the entries in the [measurements table](#).
-  **Column Auto-width.** If this button is pressed, the columns of the [measurements table](#) will continually adjust to fit the contents whenever the table changes. Click again to release the button.

### 6.4.1 Add / Edit Measurement dialog

Location: [Measurements toolbar](#) >  **Add Measurement** or  **Edit Measurement button**  
[Views menu](#) >  **Add Measurement** or  **Edit Measurement button**

Double-click a measurement in the [measurements table](#)

Purpose: allows you to add a measurement of a waveform to the selected [view](#), or edit an existing measurement



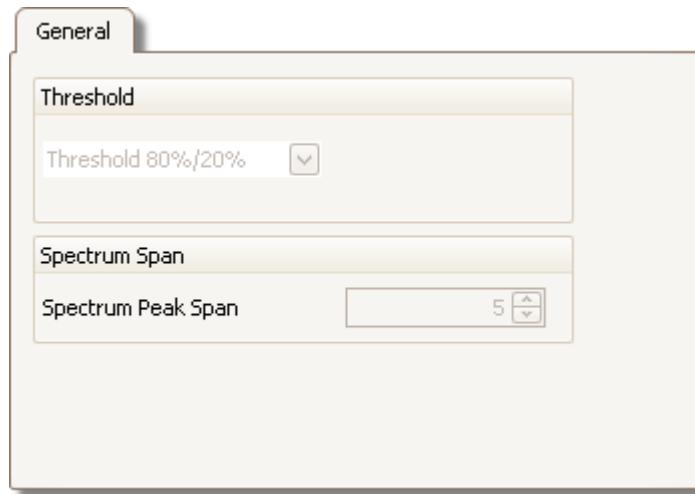
PicoScope automatically refreshes the measurement every time it updates the waveform. If this is the first measurement for the view, PicoScope will create a new [measurements table](#) to display the measurement; otherwise, it will add the new measurement to the bottom of the existing table.

<b>Channel</b>	Which of the <a href="#">scope device's</a> channels to measure.
<b>Type</b>	PicoScope can calculate a wide range of measurements for waveforms. See <a href="#">scope measurements</a> (for use with <a href="#">scope views</a> ) or <a href="#">spectrum measurements</a> (for use with <a href="#">spectrum views</a> ).
<b>Section</b>	Measure the whole trace, just the section between <a href="#">rulers</a> , or, where appropriate, a single cycle marked by one of the rulers.
<b>Advanced</b>	Gives access to <a href="#">advanced measurement settings</a> .

## 6.4.2 Advanced measurement settings

Location: [Add Measurement](#) or [Edit Measurement](#) dialog > **Advanced**

Purpose: adjusts parameters of certain measurements such as filtering and [spectrum analysis](#)

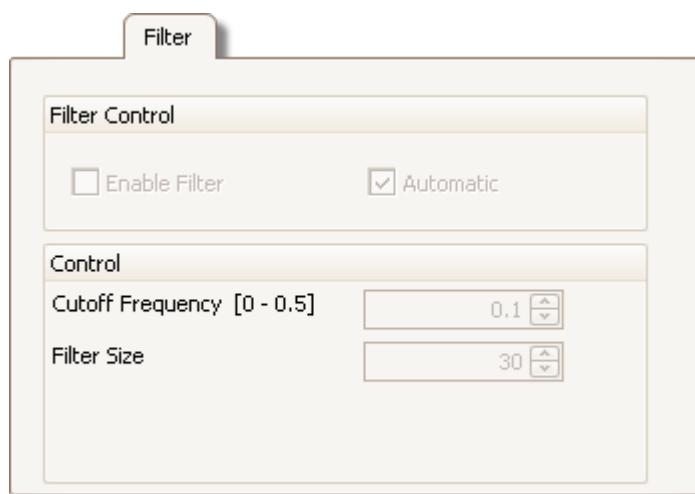


### Threshold

Some measurements, such as **Rise Time** and **Fall Time**, can be made using different thresholds. Select the appropriate ones here. When comparing rise and fall times with manufacturers' specifications, it is important to use the same thresholds for all measurements.

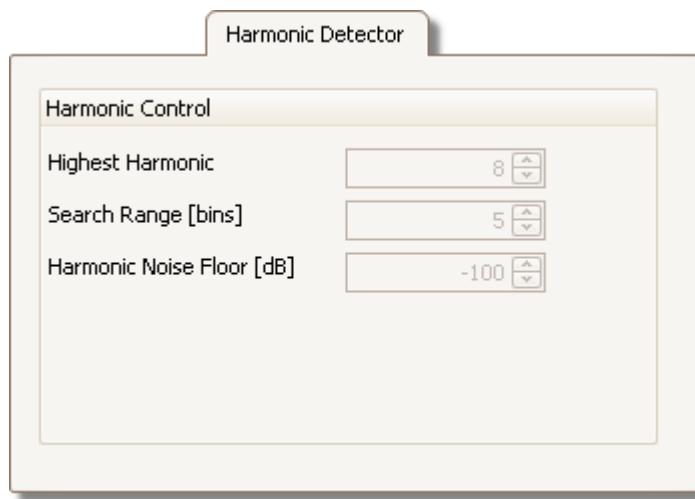
### Spectrum Span

When measuring peak-related parameters such as '[Frequency at Peak](#)' in a [spectrum view](#), PicoScope can search for a peak near to the specified [ruler](#) location. This option tells PicoScope how many frequency bins to search. The default is 5, which tell PicoScope to search from 2 bins below to 2 bins above the ruler frequency, giving a total range of 5 bins including the ruler frequency.



<b>Filter control</b>	PicoScope can low-pass filter the statistics to produce more stable and more accurate numbers. Filtering is not available on all measurement types. <b>Enable Filter</b> - check to enable low-pass filtering, if available. An "F" will appear after the measurement name in the <a href="#">measurements table</a> . <b>Automatic</b> - check to set the low-pass filter characteristics automatically
<b>Cutoff Frequency</b>	The filter cut-off frequency normalised to the measurement rate. Range: 0 to 0.5.

<b>Filter Size</b>	The number of samples used to construct the filter
--------------------	--

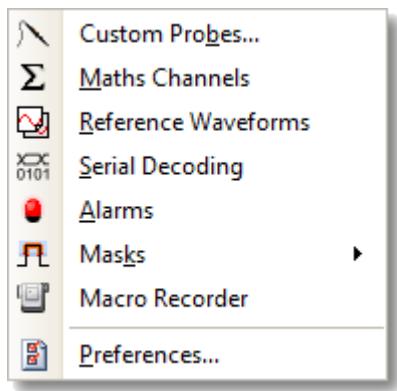


<b>Harmonic Control</b>	These options apply to distortion measurements in <a href="#">spectrum views</a> . You can specify which harmonics PicoScope uses for these measurements.
<b>Highest Harmonic</b>	The highest harmonic to include when calculating distortion power
<b>Search Range</b>	The number of frequency bins to search, centred on the expected frequency, when looking for a harmonic peak
<b>Harmonic Noise Floor</b>	The level in dB above which signal peaks will be counted as harmonics

## 6.5 Tools menu

Location: [Menu bar](#) > **Tools**

Purpose: gives access to assorted tools for signal analysis



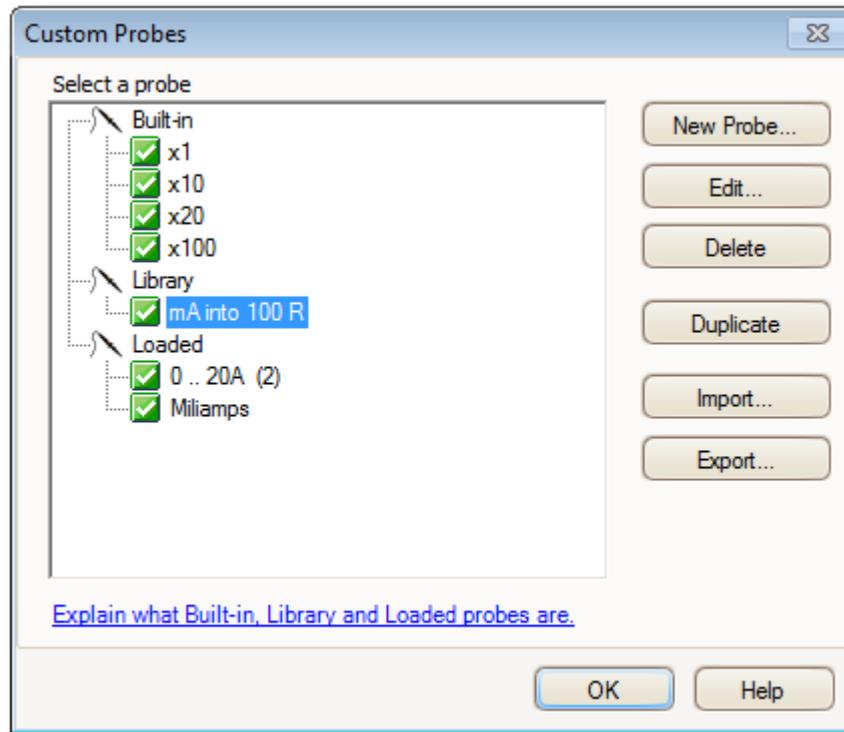
-  [\*\*Custom Probes\*\*](#): Define new probes and copy, delete, move and edit existing ones.
-  [\*\*Math Channels\*\*](#): Add or edit a channel that is a mathematical function of one or more other channels.
-  [\*\*Reference Waveforms\*\*](#): Create, load or save a channel as a copy of an existing channel.
-  [\*\*Serial Decoding\*\*](#): Decode and display the contents of serial data stream such as CAN bus.
-  [\*\*Alarms\*\*](#): Specify actions to be taken on certain events.
-  [\*\*Masks\*\*](#): Perform [mask limit testing](#) on a waveform. This detect when the waveform departs from a specified shape.
-  [\*\*Macro Recorder\*\*](#): Save a frequently-used sequence of operations.
-  [\*\*Preferences\*\*](#): Set various options that control PicoScope's behavior.

### 6.5.1 Custom Probes dialog

Location: **Tools > Custom Probes**,

or click the **Channel Options** button: 

Purpose: allows you to select predefined probes and set up [custom probes](#)



The selection of probes shown may vary depending on the version of the PicoScope software that you are using.

#### Understanding the probe list

All the probes that PicoScope knows about are listed under three main headings: **Built-in**, **Library** and **Loaded**. The probe list is preserved between sessions, so that PicoScope will never forget your custom probes unless you delete them.

- **Built-in probes.** The built-in probes are supplied by Pico Technology and do not change unless you download an authorised update from us. As a safeguard, PicoScope does not allow you to edit or delete these probes. If you want to modify one of them, you can copy it to your library by clicking **Duplicate**, and then edit the copy in your library.
- **Library probes.** These are the probes that you have created using any of the methods described in this topic. You can edit, delete or duplicate any of these probes by clicking the appropriate button in this dialog.
- **Loaded probes.** Probes in PicoScope data files ([.psdata](#)) or settings files ([.pssettings](#)) that you have opened appear here until you copy them to your library. You cannot edit or delete these probes directly, but you can click **Duplicate** to copy them to your library where you can edit them. You can also import probes from the custom ranges stored in PicoScope 5 [.psd](#) and [.pss](#) files, but these lack some of the features provided by PicoScope 6. (See "[Upgrading from PicoScope 5](#)" for more details.)

## Adding a new probe to your library

There are three ways to create a new probe:

1. Use the **Duplicate** button as described above.
2. Click **New Probe...** to define a new probe.
3. Click **Import** to load a probe definition from a `*.psprobe` file and add it to your library. These files are normally supplied by Pico, but you can also create your own by defining a new probe and then clicking **Export**.

Methods 2 and 3 open the [Custom Probe Wizard](#) to guide you through the probe definition process.

### 6.5.1.1 Custom Probe wizard

Location: [Custom Probes dialog](#) > **New Probe**

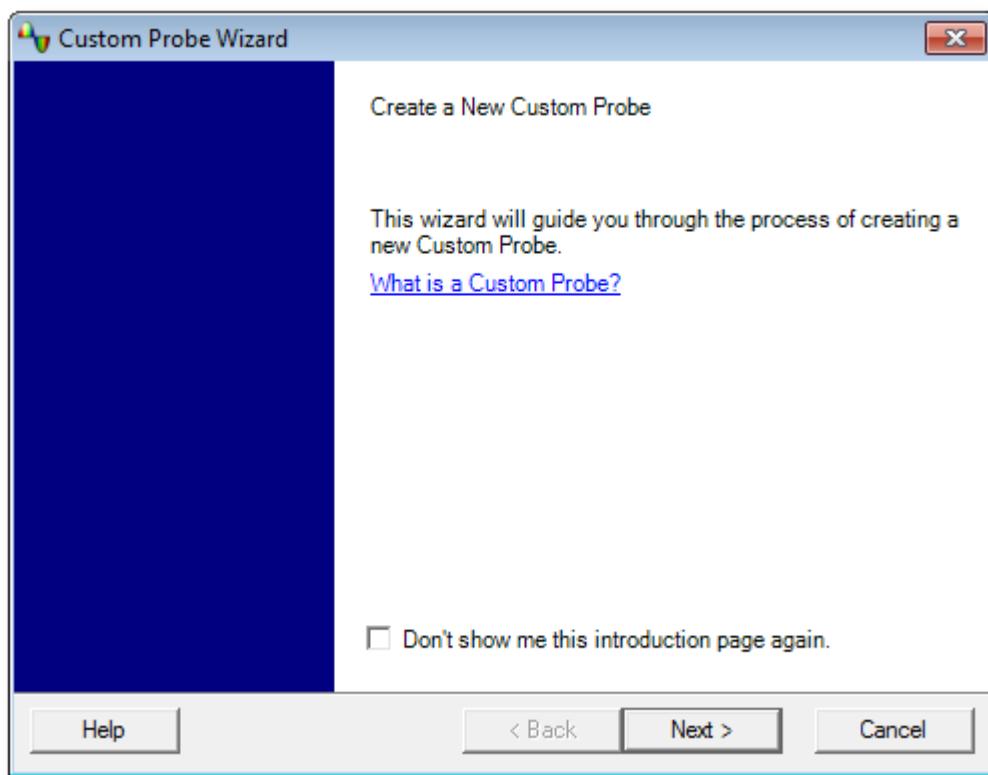
Purpose: allows you to define [custom probes](#) and set up custom ranges

The first dialog in the series is either the [Create a new Custom Probe dialog](#) or the [Edit an existing Custom Probe dialog](#).

#### 6.5.1.1.1 Create New Custom Probe dialog

Location: [Custom Probes dialog](#) > **New Probe**

Purpose: introduces you to the process for creating a new custom probe



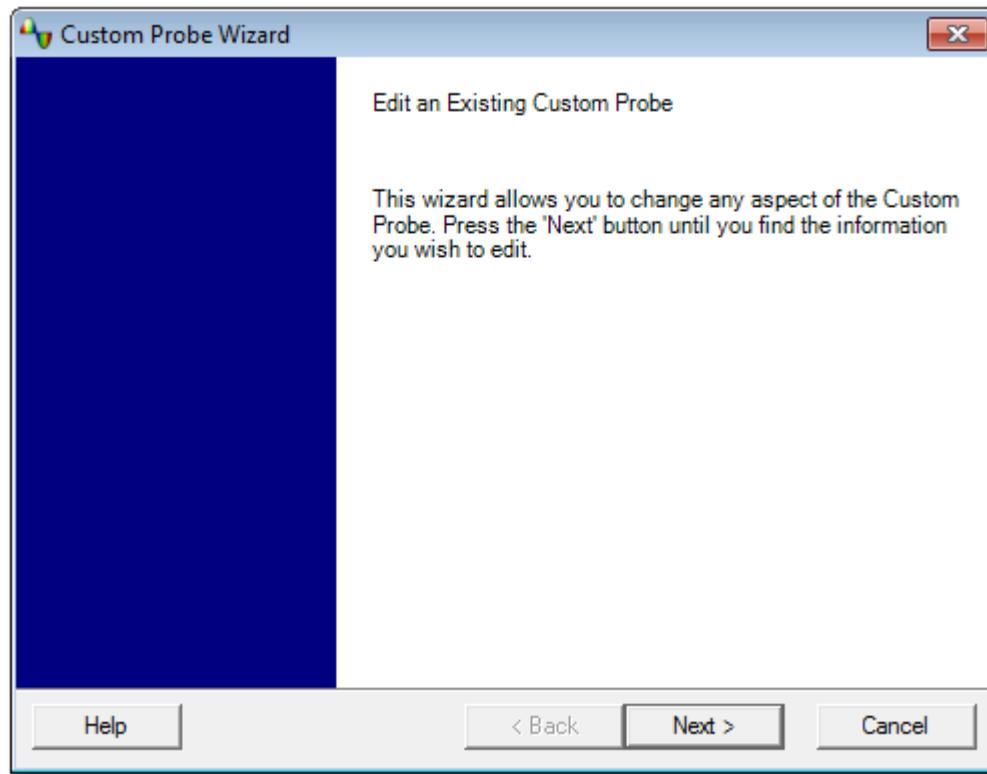
## How to use the dialog

Click **Next** to continue to the [Probe Output Units dialog](#).

6.5.1.1.2 Edit Existing Custom Probe dialog

Location: [\*\*Custom Probes dialog\*\*](#) > **Edit**

Purpose: introduces you to the process for editing an existing [custom probe](#)

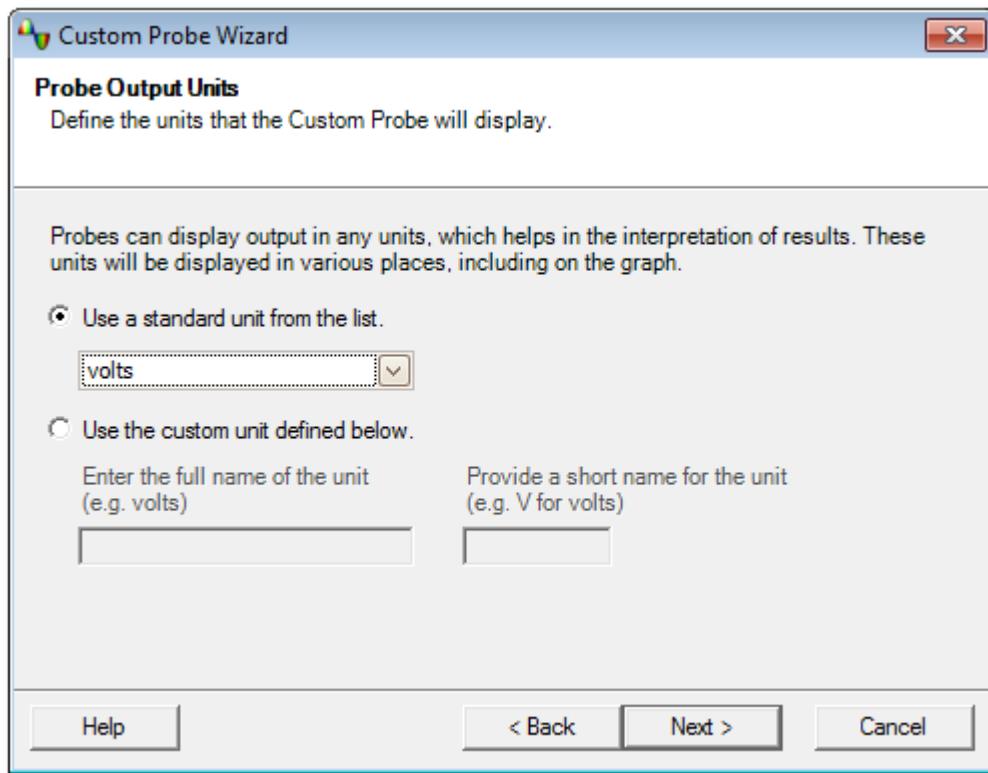


### How to use the dialog

Click **Next** to continue to the [Probe Output Units dialog](#), where you can edit the custom probe.

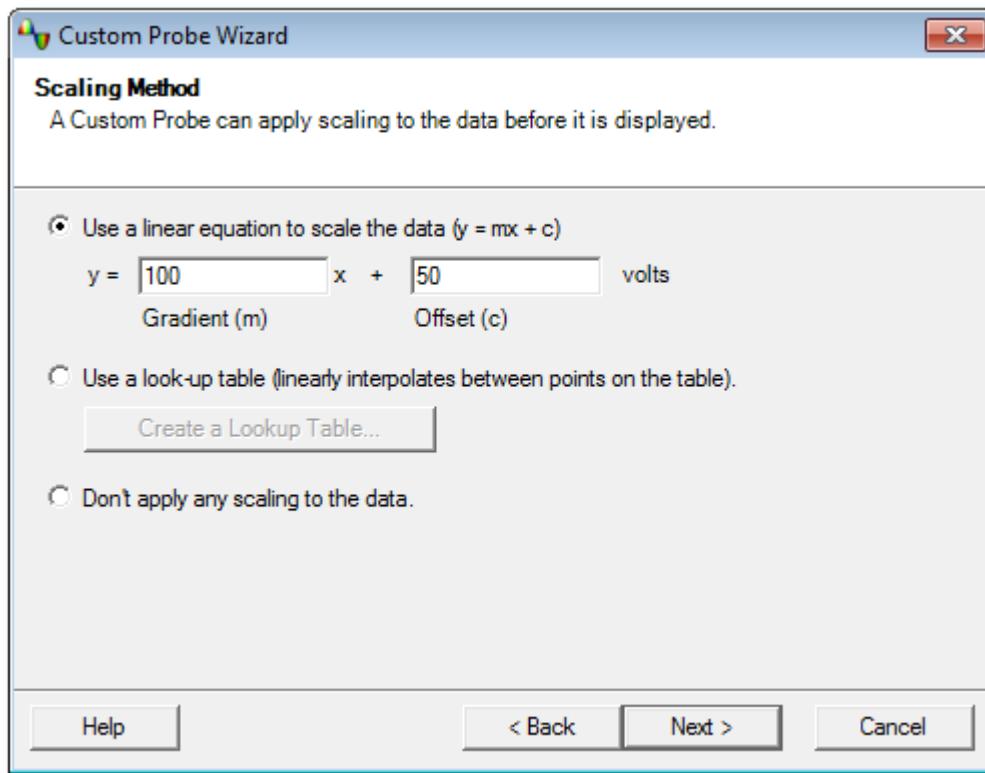
Click **Jump forward...** if you have already set up the custom probe's basic characteristics and want to add or change a custom range manually.

## 6.5.1.1.3 Probe Output Units dialog

Location: [Create new Custom Probe dialog](#) > **Next**Purpose: allows you to choose the units that PicoScope will use to display the output of your [custom probe](#)**How to use the dialog**

- ➊ To choose a standard SI unit, click **Use a standard unit from the list** and select one from the list.
- ➋ To enter a custom unit, click **Use the custom unit defined below** and type the unit name and symbol.
- ➌ Click **Next** to continue to the [Scaling Method dialog](#).
- ➍ Click **Back** to return to the [Create New Custom Probe dialog](#) if this is a new probe, or the [Edit Existing Custom Probe dialog](#) if this is an existing probe.

## 6.5.1.1.4 Scaling Method dialog

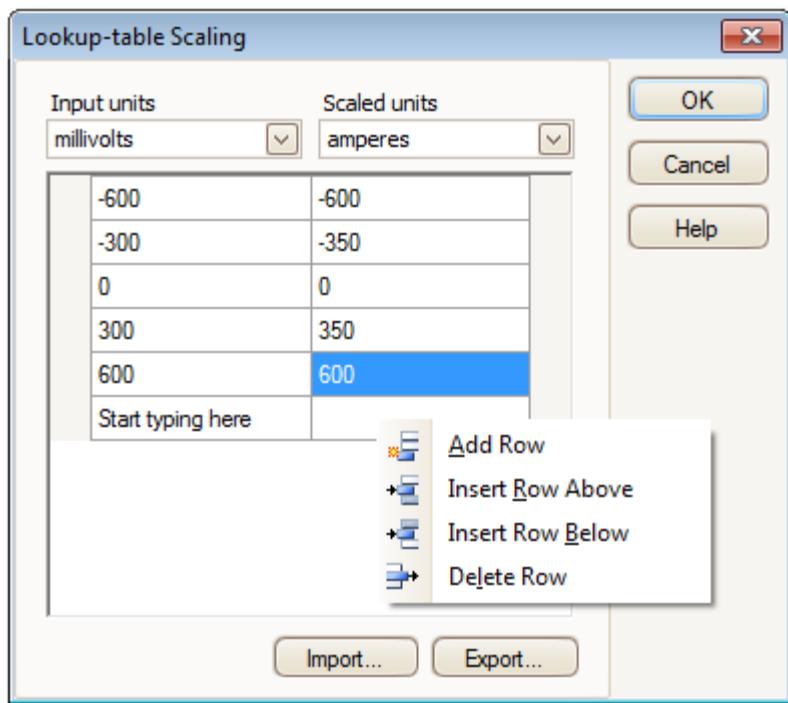
Location: [Probe Output Units dialog](#) > **Next**Purpose: allows you to define the characteristic that PicoScope will use to convert the [custom probe](#)'s voltage output to a measurement on the display**How to use the dialog**

- ➊ If you do not require any scaling or offset, click the **Don't apply any scaling** button.
- ➋ If the probe requires linear scaling, click the **Use a linear equation** button and enter the gradient (or scale factor)  $m$  and the offset  $c$  in the equation  $y = mx + c$ , where  $y$  is the displayed value and  $x$  is the probe's voltage output.
- ➌ If you wish to apply a nonlinear function to the probe's output, choose **Use a look-up table...**, then click the **Create a Lookup Table...** button to create a new lookup table. This will take you to the [Lookup-table Scaling dialog](#).
- ➍ Click **Next** to continue to the [Range Management dialog](#).
- ➎ Click **Back** to return to the [Probe Output Units dialog](#).

## 6.5.1.4.1 Lookup-table Scaling dialog

Location: [Scaling Method dialog](#) > **Create a Look-up Table or Edit the Lookup Table...**

Purpose: creates a look-up table to calibrate a [custom probe](#)



### Editing the Look-up Table

First, select suitable values in the **Input units** and **Scaled units** drop-down boxes. For example, if your probe is a current clamp that outputs one millivolt per ampere over the range -600 to +600 amperes, select **Input units** of millivolts and **Output units** of amperes.

Next, enter some data in the scaling table. Click the first empty cell at the top of the table and type "-600", then hit the **Tab** key and type "-600". When you are ready to enter the next pair of values, press the **Tab** key again to start a new row. You can also right-click on the table to obtain a more detailed menu of options, as shown in the picture. In the example above, we have entered a slightly nonlinear response; if the response had been linear then it would have been easier to use the linear option in the [Scaling Method Dialog](#).

### Import/Export

Using the **Import** and **Export** buttons, you can fill the look-up table from data in a comma-separated or tab-delimited text file, and save the look-up table to new file.

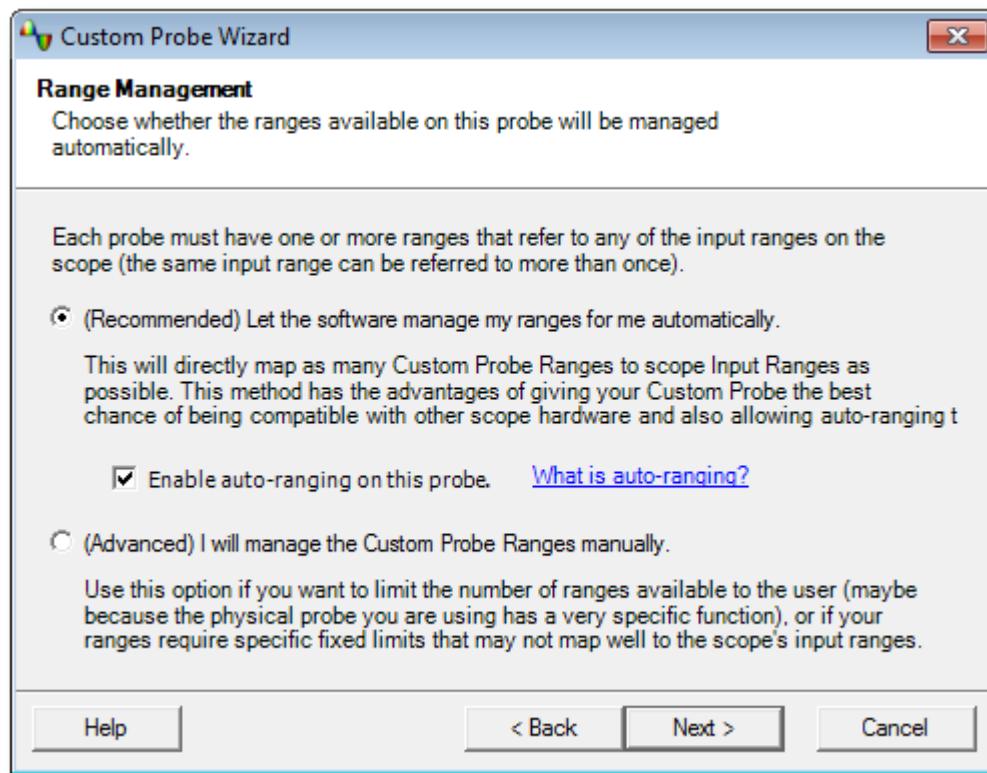
### Finishing

Clicking **OK** or **Cancel** will return you to the [Scaling Method dialog](#).

## 6.5.1.1.5 Range Management dialog

Location: **[Scaling Method dialog](#)** > **Next**

Purpose: allows you to override PicoScope's automatic range-creation feature for custom probes. In most cases, the automatic procedure will be sufficient.

**How to use the dialog**

- If you select **Let the software manage my ranges for me automatically**, then clicking **Next** will take you to the [Custom Probe Identification dialog](#). PicoScope's automatic ranges should be ideal for most applications.
- If you select **I will manage the Custom Probe Ranges manually**, clicking **Next** will take you to the [Manual Ranges Setup dialog](#).
- Click **Back** to return to the [Scaling Method dialog](#).

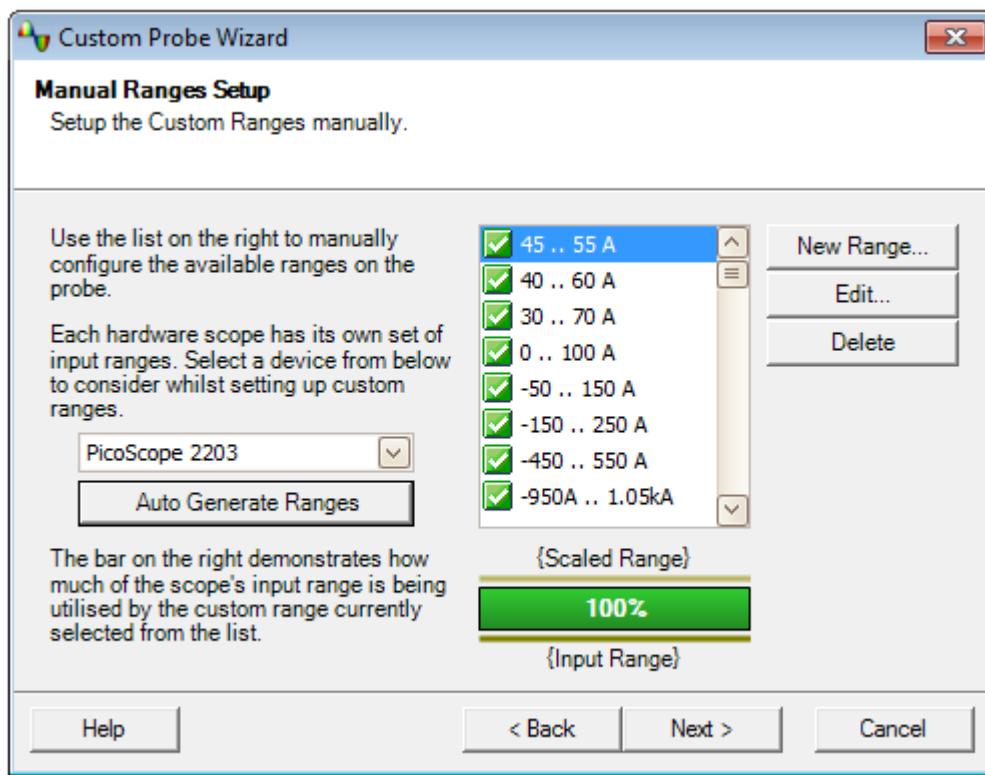
**What is Auto-ranging?**

When the **Auto-ranging** function is selected, PicoScope continually monitors the input signal and adjusts the range when necessary to allow it to display the signal with maximum resolution. This function is available on all standard ranges, and can be used with custom ranges only if you select **Let the software manage my ranges for me automatically** in this dialog.

6.5.1.1.6 Manual Ranges Setup dialog

Location: [Range Management dialog](#) > **Advanced** > **Next**

Purpose: creates ranges manually for your [custom probe](#)



### How to use the dialog

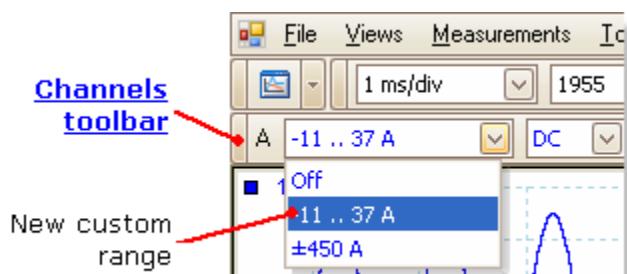
If you wish, you can click **Auto Generate Ranges** and the program will create a number of ranges for the selected device. This will create the same list of ranges that you would have obtained by selecting **Let the software manage my ranges for me automatically** in the previous dialog. When you select a range, a diagram below the list will show its relationship to the scope device's input range — this is explained further under [Edit range dialog](#). You can then edit the ranges by clicking **Edit**, or you can also add a new range by clicking **New Range**. Both of these buttons take you to the [Edit Range dialog](#).

Click **Next** to continue to the [Filter Method dialog](#).

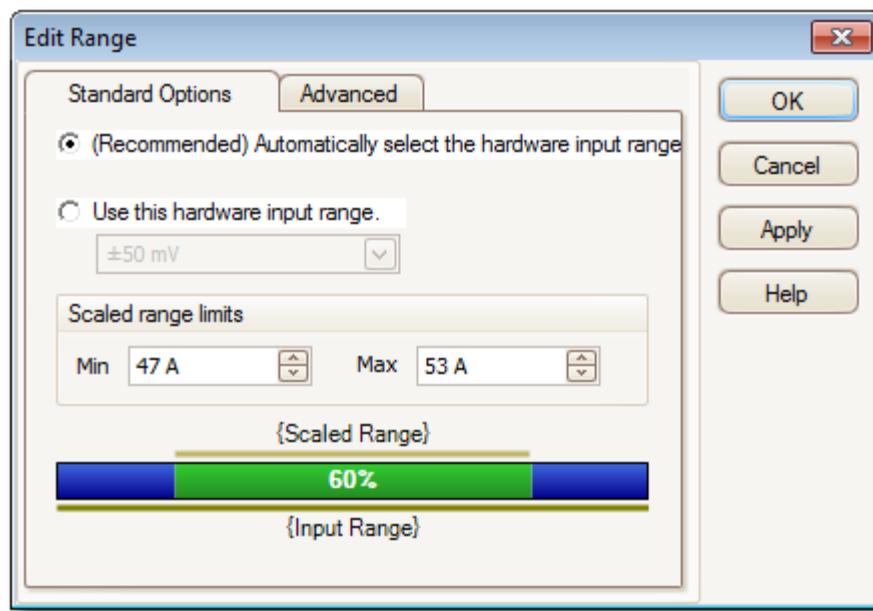
Click **Back** to return to the [Range Management dialog](#).

### How to use a new custom range

After you have created a custom range, it will appear in the drop-down list of ranges in the [Channels toolbar](#), like this:



## 6.5.1.1.6.1 Edit Range dialog

Location: [Manual Ranges Setup dialog](#) > **Edit or New Range**Purpose: editing a manual range for a [custom probe](#)**Automatic mode**

If you leave the "**Automatic**" radio button pressed, the program will automatically determine the best hardware input range for the device as you change the **Scaled range limits**. This is the best mode to use for almost all ranges. You should set the **Scaled range limits** to the maximum and minimum values you wish to see on the vertical axis of the scope display.

**Fixed range mode**

If you press the "**Hardware input range**" radio button and select a hardware input range from the drop-down box, PicoScope will then use that hardware input range whatever scaled range limits you choose. Set the upper and lower scaled range limits to the limits you wish to appear at the top and bottom of the vertical axis in PicoScope's [scope view](#).

**What is an input range?**

An input range is the signal range, usually in volts, on the input channel of the [scope device](#). Your scaled range should match this as closely as possible to make the most of the scope's resolution.

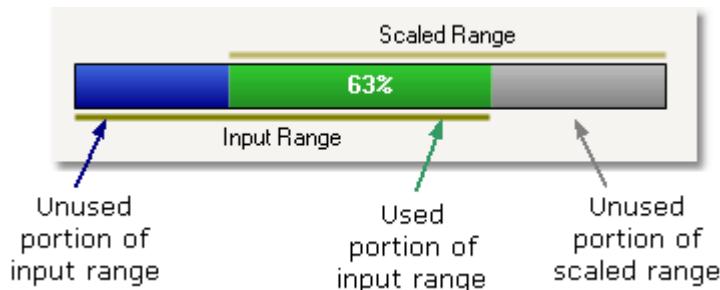
**What is a scaled range?**

The scaled range is the range that will appear on the vertical axis of the scope display when the probe is selected.

The scaling that you chose on the [Scaling Method](#) page defines the relationship between the input range and the scaled range. This dialog enables you to set up ranges to display the scaled data on the scope view.

### The range utilisation bar

This diagram at the bottom of the dialog shows how well the input range of the device is matched to the scaled range.



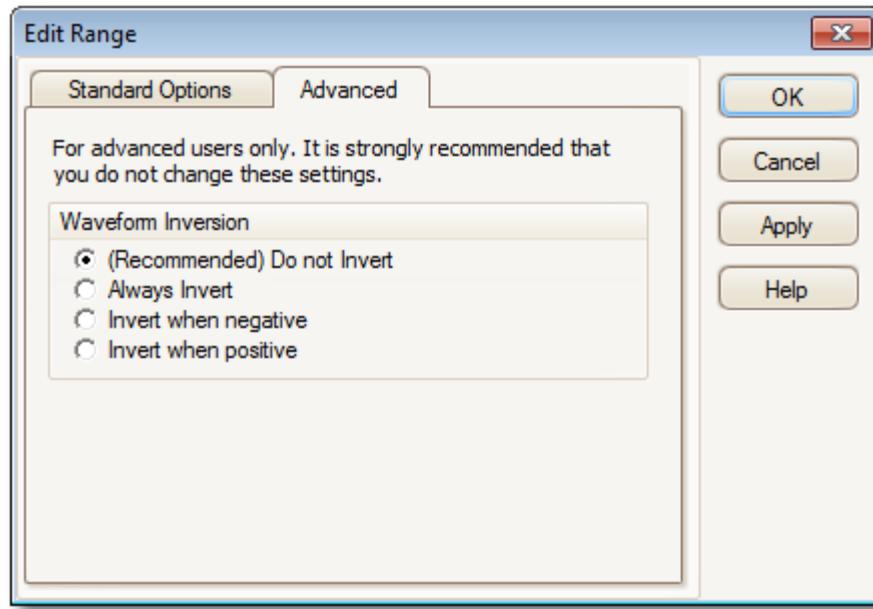
- **Green** - The section of the input range that is used by the scaled range. This should be as large as possible, to maximise the use of the scope device's resolution.
- **Blue** - Areas of the input range that are not being used. These indicate wasted resolution.
- **Grey** - Parts of the scaled range that are not covered by the input range. These will result in wasted space on the graph. The range utilisation bar may not represent these areas accurately when non-linear scaling is being used, so you should always test the scaled range limits on the scope view.

### Advanced tab

#### Finishing

Clicking **OK** or **Cancel** will return you to the [Manual Ranges Setup dialog](#).

## 6.5.1.1.6.2 Edit Range dialog (Advanced tab)

Location: [Manual Ranges Setup dialog](#) > **Edit** or **New Range** > **Advanced** tabPurpose: configuring advanced options for [custom probes](#)

These options are for factory use and we recommend that you do not change them.

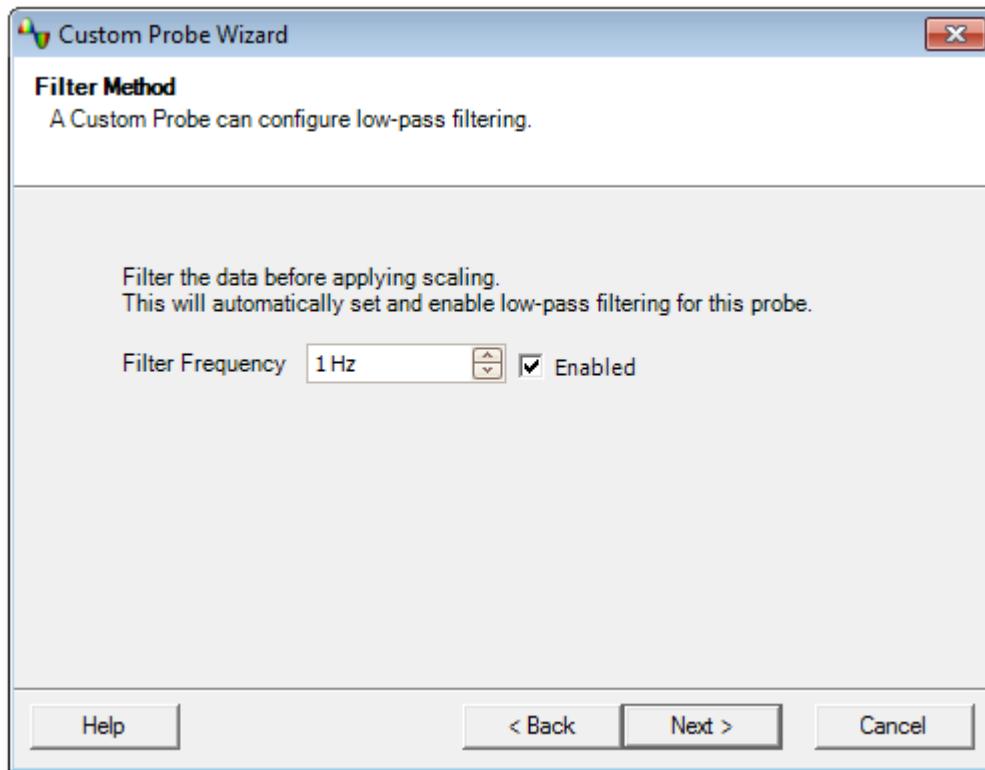
## Finishing

Clicking **OK** or **Cancel** will return you to the [Manual Ranges Setup dialog](#).

6.5.1.1.7 Filter Method dialog

Location: [\*\*Manual Ranges Setup dialog\*\*](#) > **Next**

Purpose: sets up lowpass filtering for this custom probe



This dialog has the same effect as manually enabling the [Lowpass Filtering](#) option in the [Channel Options dialog](#). Filtering will only occur if the attached scope device supports filtering.

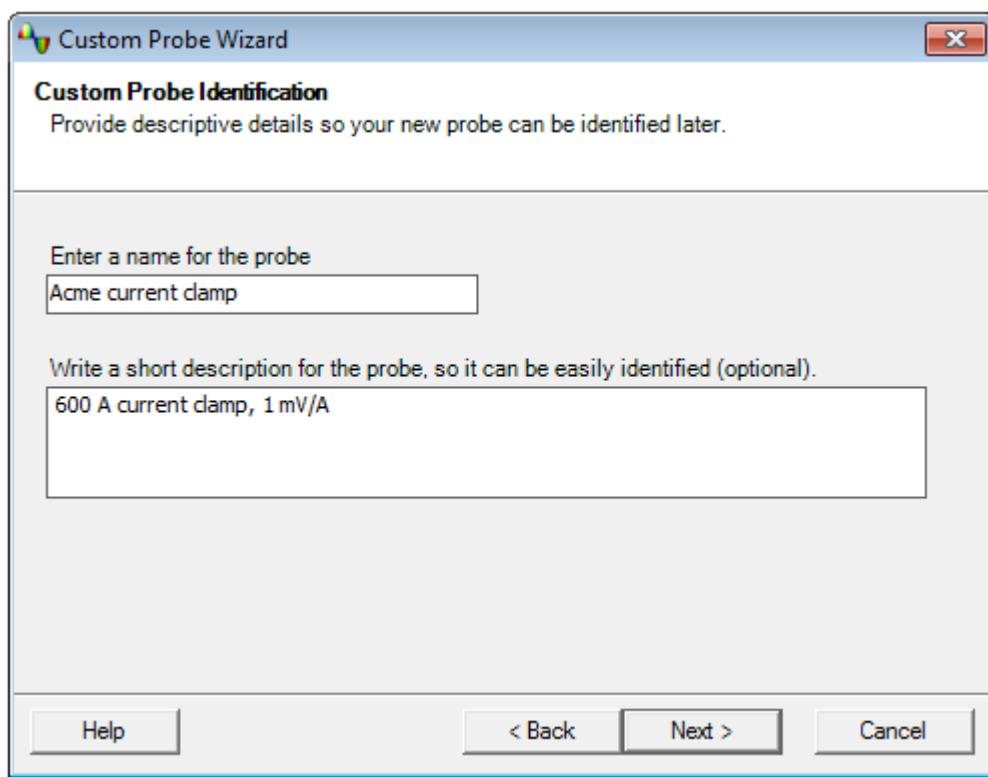
**Back:** Go to the [Manual Range Setup dialog](#)

**Next:** Go to the [Custom Probe Identification dialog](#)

6.5.1.1.8 Custom Probe Identification dialog

Location: [Range Management dialog](#) > **Next**

Purpose: entering text to identify the [custom probe](#)



### How to use the dialog

Click **Back** to return to the [Filter Method dialog](#).

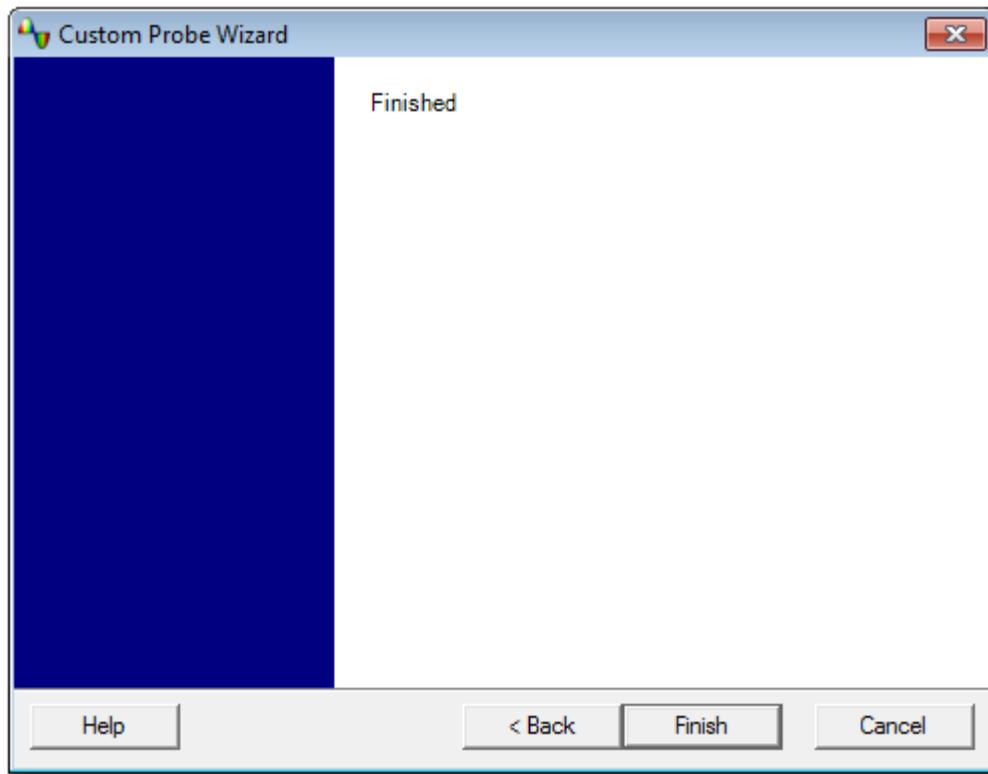
- The **probe name** will appear in the probe list.
- The **description** is not used in the present version of the software.

Fill in the text fields and click **Next** to continue to the [Custom Probe Finished dialog](#).

6.5.1.1.9 Custom Probe Finished dialog

Location: [\*\*Custom Probe Identification dialog\*\*](#) > **Next**

Purpose: signals the end of the [custom probe](#) setup procedure



### How to use the dialog

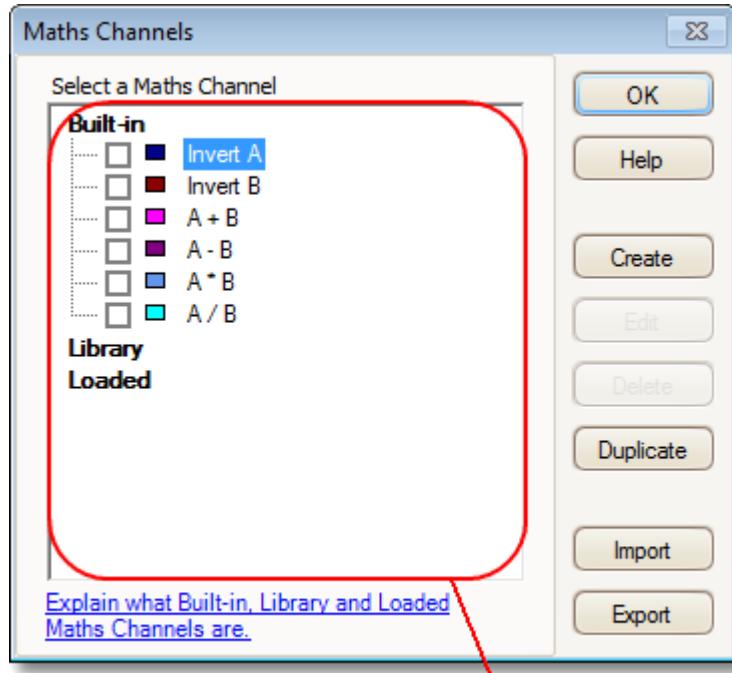
Click **Back** to return to the [Custom Probe Identification dialog](#).

Click **Finish** to accept your custom probe settings and return to the [Custom Probes dialog](#).

### 6.5.2 Math Channels dialog

Location: [Tools > Math Channels](#)

Purpose: [creating](#), [editing](#) and controlling [math channels](#), which are virtual channels generated by mathematical functions of input channels



**Math Channel list**

**Math Channel list** The main area of the **Math Channels dialog** is the **Math Channel list**, which shows all the built-in, library and loaded [math channels](#). To choose whether or not a channel appears in the main [PicoScope window](#), click the appropriate check box and then **OK**. You can have up to 8 channels in any view, including input channels and math channels. If you enable a 9th channel, PicoScope opens a new view.

**Built In:** these math channels are defined by PicoScope and cannot be changed

**Library:** these are the math channels that you define using the **Create** or **Duplicate** button, **Edit**, or load with the **Import** button

**Loaded:** these are the math channels present in any PicoScope settings or data files that you have loaded

#### Create

Opens the [Math Channel Wizard](#), which guides you through the process of creating or editing a math channel. The new channel will appear under "**Library**" in the **Math Channel list**.

<b>Edit</b>	Opens the <b>Math Channel Wizard</b> to allow you to edit the selected math channel. You must first select a channel in the <b>Library</b> section of the <b>Math Channel list</b> . If the channel you want to edit is in the <b>Built In</b> or <b>Loaded</b> section, first copy it to the <b>Library</b> section by clicking <b>Duplicate</b> , then select it and click <b>Edit</b> .
<b>Delete</b>	Permanently deletes the selected math channel. Only math channels in the <b>Library</b> section can be deleted.
<b>Duplicate</b>	Creates a copy of the selected math channel. The copy is placed in the <b>Library</b> section, from where you can edit it by clicking <b>Edit</b> .
<b>Import</b>	Opens a <code>.psmaths</code> math channel file and places the math channels it contains in the <b>Library</b> section.
<b>Export</b>	Saves all math channels from the <b>Library</b> section to a new <code>.psmaths</code> file.

### 6.5.2.1 Math Channel Wizard

Location: **[Channel Setup toolbar](#) > [Math Channels button](#)**

Purpose: creating, editing and controlling math channels, which are virtual channels generated by mathematical functions of input channels



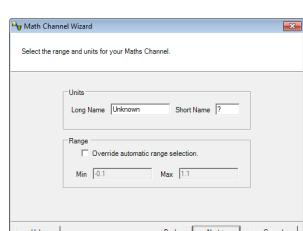
#### 1. [Introduction](#)



#### 2. [Equation](#)



#### 3. [Channel name](#)



#### 4. [Units and range](#)

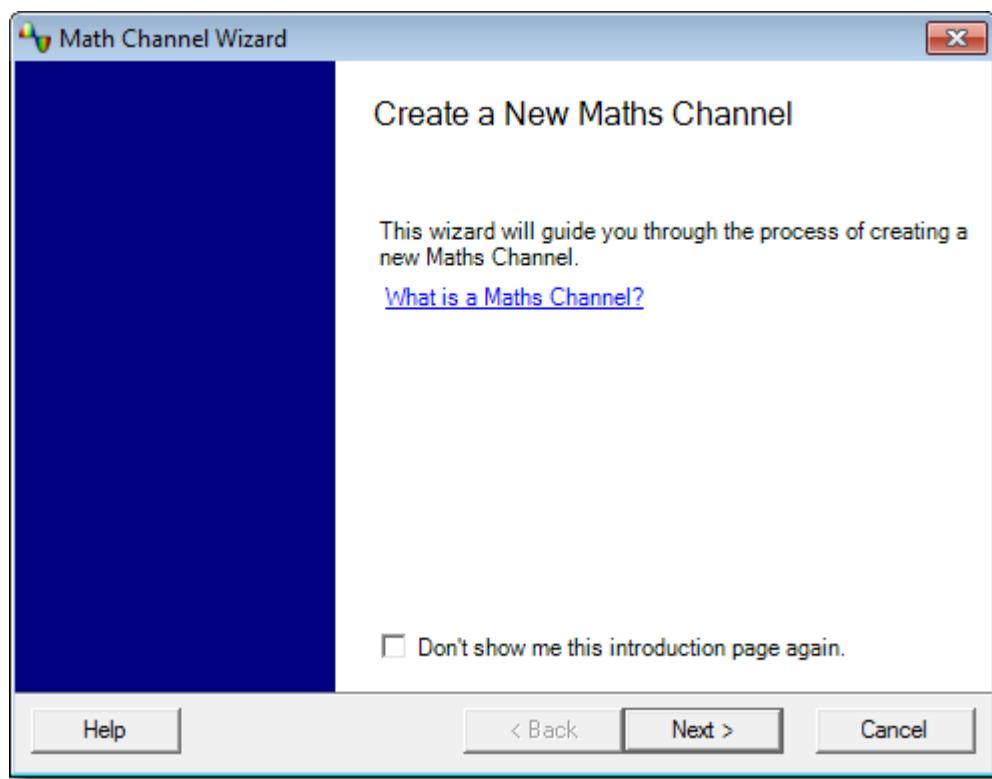


#### 5. [Finished](#)

6.5.2.1.1 Math Channel Wizard Introduction dialog

Location: **[Math Channels dialog](#)** > **Create** (if you have not ticked the "Don't show me this introduction page again" check box)

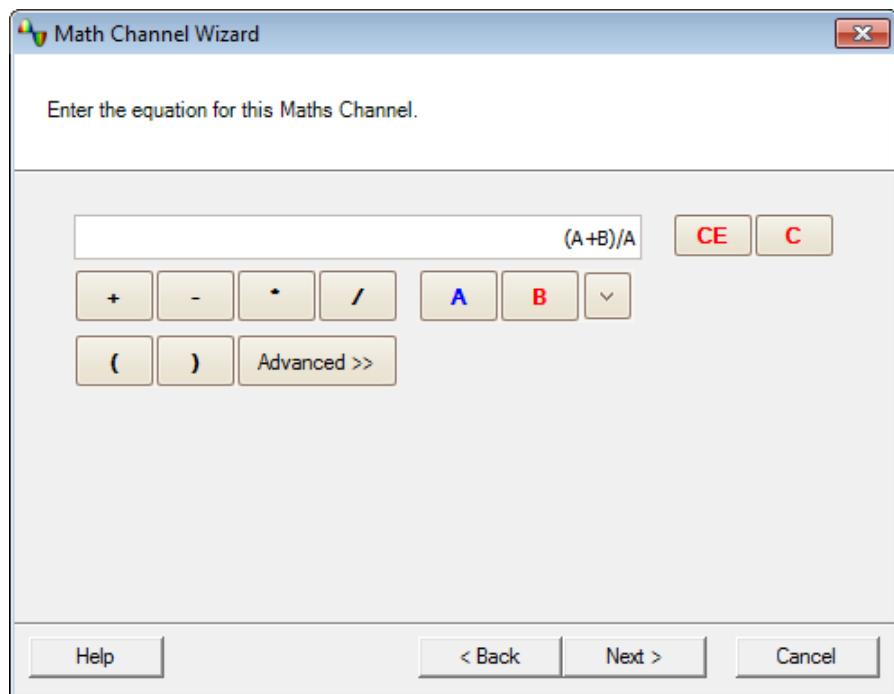
Purpose: introduces the **[Math Channel Wizard](#)**



## 6.5.2.1.2 Math Channel Wizard Equation dialog

Location: [\*\*Math Channel Wizard\*\*](#)

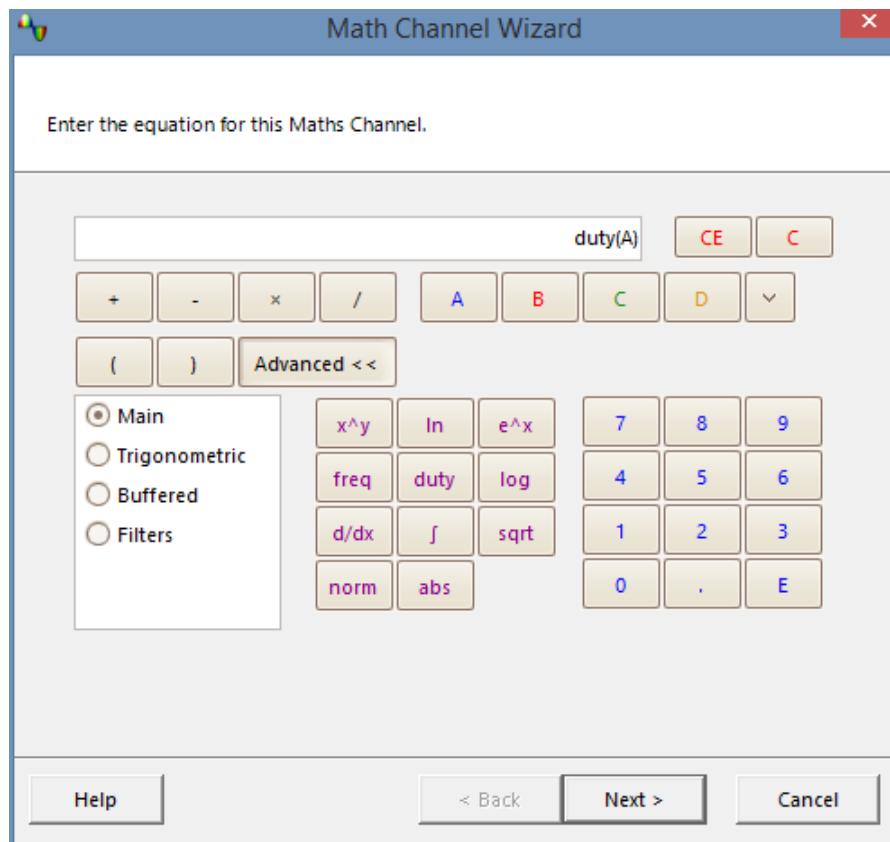
Purpose: allows you to enter or edit the equation for a [math channel](#). You can type directly into the equation box, or click the calculator buttons and let the program insert the symbols for you. A red error indicator  will appear to the right of the equation box if the equation contains a syntax error.

**Basic view****Math Channel Wizard Equation dialog, basic view****Basic buttons**

Button	Equation	Description
		<b>Clear Equation.</b> Clears the entire contents of the equation box.
		<b>Clear.</b> Clears the single character to the left of the cursor.
	+	<b>Add</b>
	-	<b>Subtract</b> (or negate)
	*	<b>Multiply</b>
	/	<b>Divide</b>
 ... 	A...D	<b>Input Channels.</b> The selection available depends on the number of enabled channels on the scope device.
	{...}, T	<b>Other Operands.</b> Shows a drop-down list of available inputs to equations, including <a href="#">reference waveforms</a> and <a href="#">Time</a> .
	(...)	<b>Brackets.</b> Expressions within brackets will be evaluated before the expressions to either side.

## Advanced view

Clicking the **Advanced** button reveals more function buttons. Initially these are the **Main** group of buttons, as shown below.



Equation Editor Advanced view, showing Main buttons

### Advanced buttons (numeric keypad)

Button	Equation	Description
0	0..9	<b>0 to 9.</b> The decimal digits.
.	.	<b>Decimal point</b>
E	E	<b>Exponent.</b> $aEb$ means $a \times 10^b$ .

### Advanced buttons (Main group)

Button	Equation	Description
sqrt	sqrt()	<b>Square root</b>
x^y	$x^y$	<b>Power.</b> Raise $x$ to the power of $y$ .
ln	ln()	<b>Natural logarithm</b>
abs	abs()	<b>Absolute value</b>
freq	freq()	<b>Frequency.</b> Calculated in hertz.
norm	norm()	<b>Normalise.</b> PicoScope calculates the maximum and minimum values of the argument over the capture period, and then scales and offsets the argument so that it exactly fits the range [0, +1] units.
e^x	exp()	<b>Natural exponent.</b> Raise $e$ , the base of the natural logarithm, to the power of $x$ .

	log()	<b>Logarithm.</b> Base-10 logarithm.
	derivative()	<b>Derivative.</b> Calculated with respect to the x-axis. Note: the derivative of a sampled signal contains a large amount of noise, so it is advisable to apply <a href="#">digital low-pass filtering</a> to all channels used as inputs to this function.
	integral()	<b>Integral.</b> Along the x-axis.

Click **Trigonometric**, **Buffered** or **Filters** to reveal alternative groups of buttons:

### Advanced buttons (Trigonometric group)

Button	Equation	Description
	pi	<b>Pi.</b> The ratio of a circle's circumference to its diameter.
		<b>Inverse.</b> Modifies the <b>sin</b> , <b>cos</b> and <b>tan</b> buttons to the inverse trigonometric functions <b>asin</b> , <b>acos</b> and <b>atan</b> .
	sin()	<b>Sine.</b> The operand is in radians.
	cos()	<b>Cosine.</b> The operand is in radians.
	tan()	<b>Tangent.</b> The operand is in radians.
	sinh()	<b>Hyperbolic sine.</b>
	cosh()	<b>Hyperbolic cosine.</b>
	tanh()	<b>Hyperbolic tangent.</b>

### Advanced buttons (Buffered group)

When the scope is running, these functions operate continuously on all waveforms since the scope started capturing. If a math channel containing these functions is enabled when the scope is stopped, it operates on the contents of the waveform buffer instead.

Button	Equation	Description
	min()	<b>Minimum.</b> Negative peak detect of all previous waveforms.
	max()	<b>Maximum.</b> Positive peak detect of all previous waveforms.
	average()	<b>Average.</b> Arithmetic mean of all previous waveforms.
	peak()	<b>Peak detect.</b> Display maximum-to-minimum range of all previous waveforms.

### Advanced buttons (Filters group)

Parameters:

**i** is the input channel or other operand (see under **Basic buttons** above)  
**f** (or **f<sub>1</sub>** and **f<sub>2</sub>**) are the -3 dB cutoff frequencies of the filter, in hertz

Button	Equation	Description
	HighPass( <i>i,f</i> )	<b>High pass filter.</b> Attenuates low frequencies.
	LowPass( <i>i,f</i> )	<b>Low pass filter.</b> Attenuates high frequencies.

BandPass( $i, f_1, f_2$ )BandStop( $i, f_1, f_2$ )**Band pass filter.** Attenuates high and low frequencies outside the specified range.**Band stop filter.** Attenuates mid-band frequencies inside the specified range.

These are digital filters with a finite number of taps and therefore cannot attenuate down to DC. They have a minimum cutoff frequency of 1/64 000 of the scope's sampling rate. You can find the current sampling rate displayed in the [Properties sheet](#).

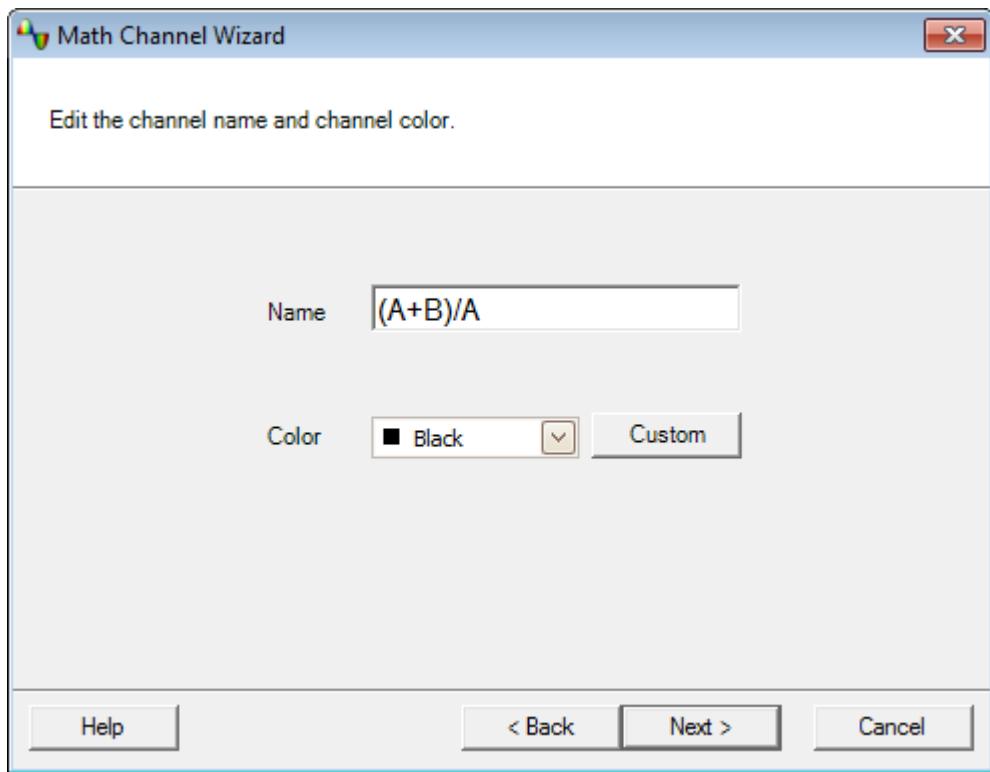
### Additional functions

There are a few operators that can be entered only by using the equation box:

**Signum function.** The **sign()** operator returns the sign of its input. The result is +1 when the input is positive, -1 when the input is negative, and 0 when the input is 0.

**Advance/Delay.** Add **[t]** after a channel name to advance it by  $t$  seconds. For example, **A[0.001]** is equal to Channel A advanced by 1 millisecond, and **A[-0.001]** is equal to Channel A delayed by 1 millisecond.

## 6.5.2.1.3 Math Channel Wizard Name dialog

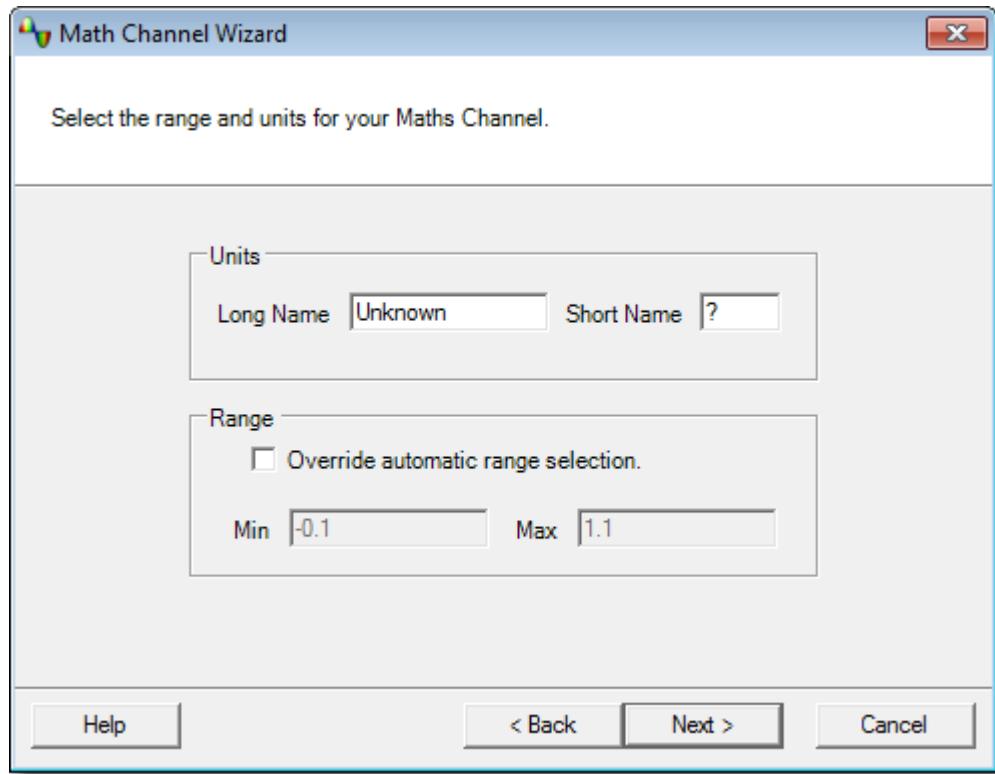
Location: [\*\*Math Channel Wizard\*\*](#)Purpose: allows you to enter or edit the name and color of a [math channel](#)

PicoScope initially sets the name to the text of the equation, but you can edit it to anything you like. The name will appear in the channel list in the [Math Channels dialog](#). You can set the color of the trace to one of the standard colors in the drop-down-list, or click **Custom** to choose any possible color allowed by Windows.

## 6.5.2.1.4 Math Channel Wizard Units and Range dialog

Location: [\*\*Math Channel Wizard\*\*](#)

Purpose: allows you to specify the measurement units and the range of values to display for a [math channel](#)



**Units, Long Name:** This is for your reference only.

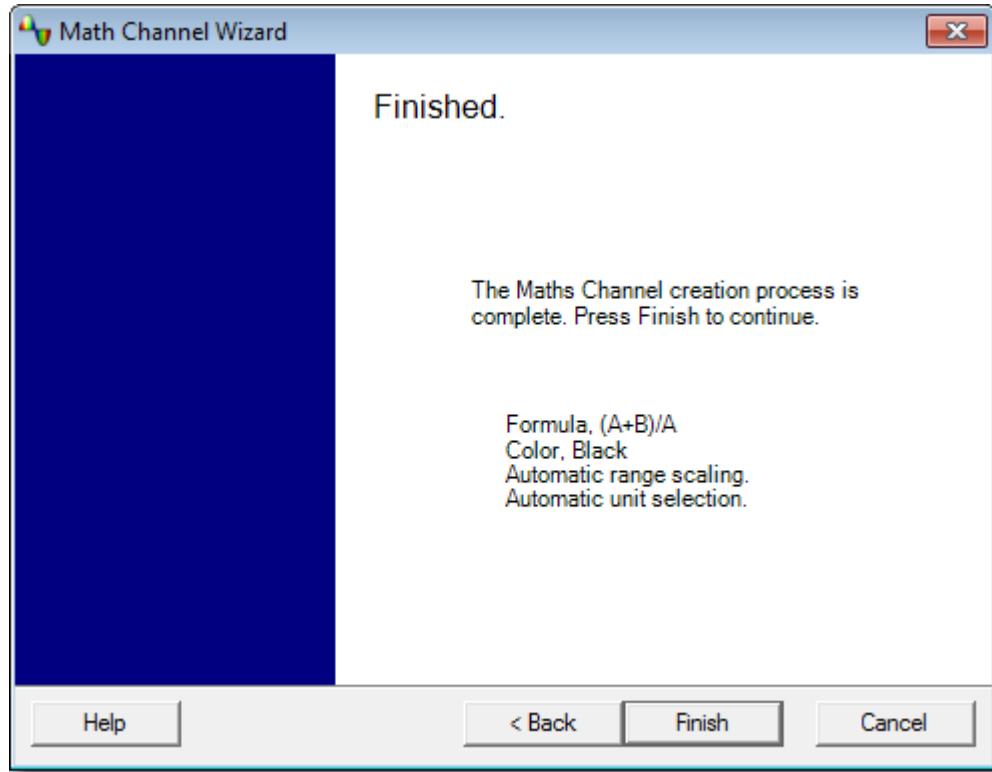
**Units, Short Name:** This will be displayed on the measurement axis in [scope](#) and [spectrum](#) views, in the [ruler legend](#) and in the [measurements table](#).

**Range:** If you leave the check box empty, PicoScope will choose the most appropriate range for the measurement axis. If you prefer to set your own values for the minimum and maximum extremes of the measurement axis, tick the check box and enter them in the **Min** and **Max** boxes.

## 6.5.2.1.5 Math Channel Wizard Finished dialog

Location: [\*\*Math Channel Wizard\*\*](#)

Purpose: shows you the settings for the [math channel](#) that you have just created or edited



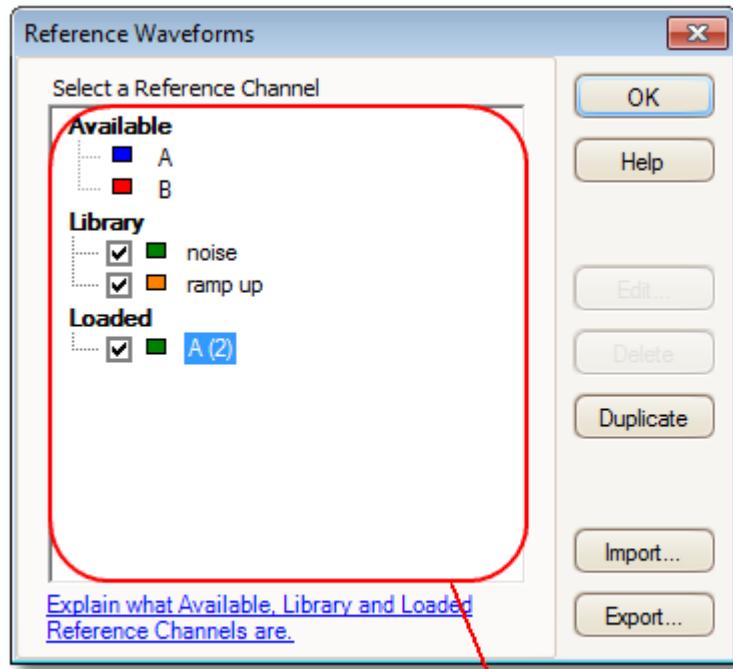
**Back.** Click this button to return to previous dialogs in the [\*\*Math Channel Wizard\*\*](#) if you wish to change any of the settings.

**Finish.** Click this button to accept the settings shown and return to the [\*\*Math Channels dialog\*\*](#). If you want the new or edited channel to appear on the scope or spectrum display, remember to tick the appropriate check box in the channel list. You can change them later by clicking the **Math Channels** button in the [\*\*Channel Setup toolbar\*\*](#).

### 6.5.3 Reference Waveforms dialog

Location: **Tools > Reference Waveforms**

Purpose: enables you to create, [edit](#) and control [reference waveforms](#), which are stored copies of input channels



Reference Waveforms list

**Reference Waveforms list** The main area of the **Reference Waveforms dialog** is the **Reference Waveforms list**, which shows all the available input channels and the library and loaded [reference waveforms](#). To choose whether or not a waveform appears in the main [PicoScope window](#), click the appropriate check box and then **OK**. You can have up to 8 channels in any view, including input channels, math channels and reference waveforms. If you try to enable a 9th channel, PicoScope will open another view.

**Available:** these input channels are suitable as sources for reference waveforms

**Library:** these are the reference waveforms that you have defined using the **Duplicate** button or loaded with the **Import** button

**Loaded:** these are the reference waveforms present in any PicoScope settings or data files that you have loaded

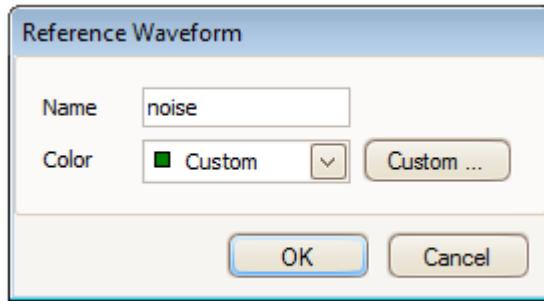
**Edit** Opens the [Edit Reference Waveform dialog](#) to allow you to edit the selected reference waveform. You must first select a waveform in the **Library** section of the **Reference Waveforms list**. If the waveform you want to edit is in the **Loaded** section, first copy it to the **Library** section by clicking **Duplicate**, then select it and click **Edit**.

<b>Delete</b>	Permanently deletes the selected reference waveform. Only reference waveforms in the <b>Library</b> section can be deleted.
<b>Duplicate</b>	Creates a copy of the selected input channel or reference waveform. The copy is placed in the <b>Library</b> section, from where you can edit it by clicking <b>Edit</b> . A quicker way to do the same thing is to right-click on the view, select <b>Reference Waveforms</b> and then click the channel that you wish to copy.
<b>Import</b>	Opens a <code>.psreference</code> reference waveform file and places the waveforms it contains in the <b>Library</b> section.
<b>Export</b>	Saves all reference waveforms from the <b>Library</b> section to a new <code>.psreference</code> or MATLAB 4 <code>.mat</code> file.

#### 6.5.3.1 Edit Reference Waveform dialog

Location: [Reference Waveforms dialog > Edit](#)

Purpose: allows you to edit the name and color of a [reference waveform](#)



**Name:** PicoScope initially names the waveform after the input channel used as its source, but you can edit it to anything you like. Here we have named it "sine". The name will appear in the waveform list in the [Reference Waveforms dialog](#).

**Color:** You can set the color of the trace to one of the standard colors in the drop-down-list, or click **Custom** to choose any color allowed by Windows.

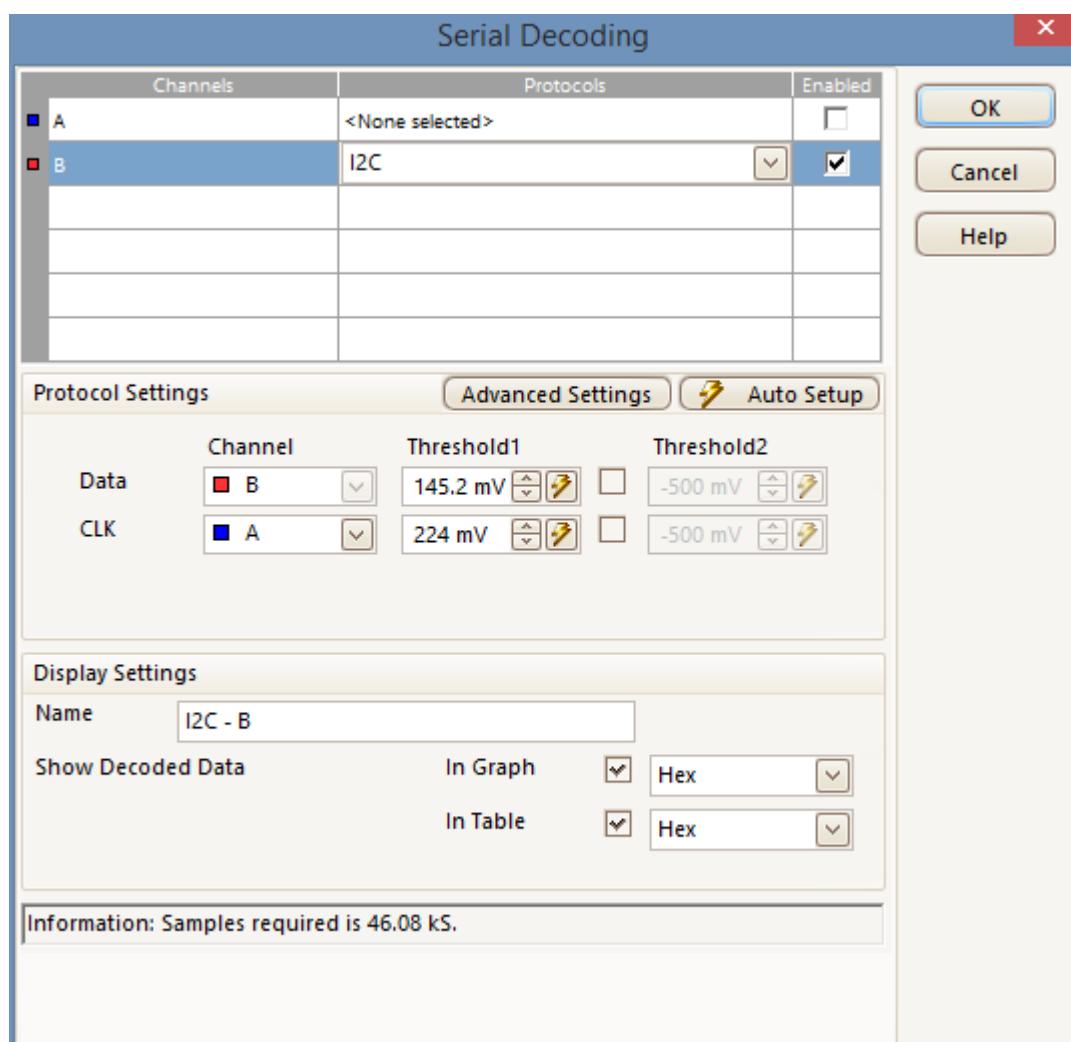
### 6.5.4 Serial Decoding dialog

Location: **Tools > Serial Decoding**

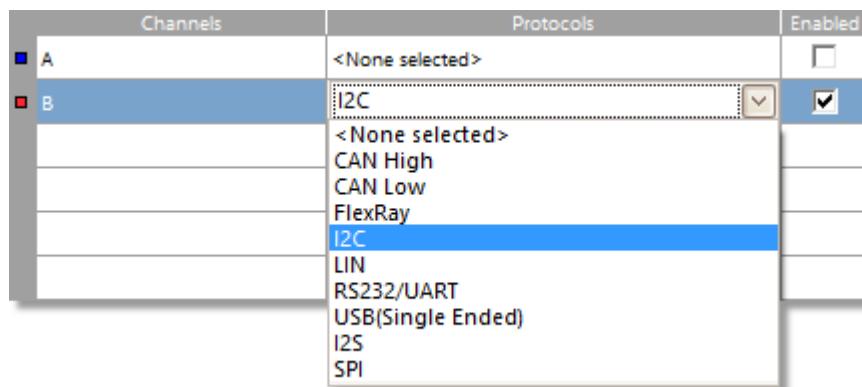
Purpose: lets you choose which channels to use for [serial decoding](#) and set other options

The following formats are supported:

- [I<sup>2</sup>C](#)
- [CAN Bus](#)
- [LIN](#)
- [FlexRay](#)
- [I<sup>2</sup>S](#)
- [RS232 \(UART\)](#)
- [SPI](#)
- USB (Single Ended)



## Protocols table



This is where you select which channels to decode, and what serial protocol to use for each channel. All the available channels are listed in the **Channels** column.

- If the channel you want to use is not listed, enable it first by using the [Channels toolbar](#).
- For each channel that you want to decode, click in the **Protocols** column. A drop-down list will appear, containing all the protocols that PicoScope understands. The list may vary depending on which version of PicoScope you are using. For multi-channel protocols like I<sup>2</sup>C and SPI, select the data channel. Any other channels will be specified later.
- Select the protocol you wish to use.
- Change the settings in the **Protocol Settings** panel if required.
- Change further settings by clicking [Advanced Settings](#) if required.
- Check the **Enabled** box for the channels that you want to decode.

## Protocol Settings

The options in this section depend on which protocol is selected.



### Controls:

[Advanced Settings](#)

Opens the [Advanced Decoder Settings](#) dialog for options that are rarely changed.

### Baud Rate.

The symbol frequency in baud. Choose one of the standard values from the drop-down list, type in an arbitrary value, or click to detect the baud rate automatically.

### Channel.

The channel attached to the specified signal. Configure this to match the connection of the inputs to the scope device.

**Threshold1.**

For single-threshold decoding, the voltage that defines the transition in either direction between high and low logic states. For dual-threshold decoding, the voltage that defines the transition from low to high logic states. To activate dual-threshold decoding, check the check box to the left of the **Threshold2** control.

**Threshold2.**

The voltage that defines the transition from high to low logic states. Used only with dual-threshold decoding. To activate dual-threshold decoding, check the check box to the left of the **Threshold2** control.

**Signals:****I/O, Data or D.**

Data channel for various protocols. Bit order, number of bits and polarity are specified in [Advanced Decoder Settings](#).

Note: bidirectional SPI can be decoded as two separate buses, one with Data In (DI) as the I/O signal and the other with Data Out (DO) as the I/O signal. The Clk (and Chip Sel if present) can be shared by both buses.

**Clk, CLK or Clock.**

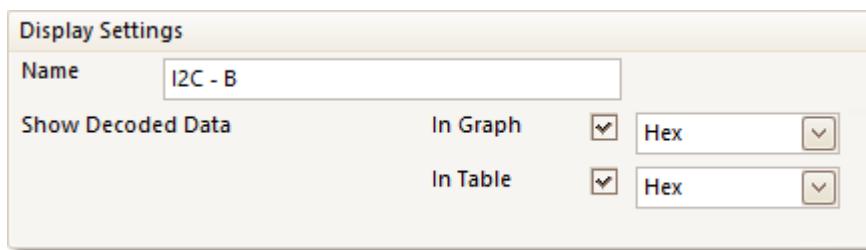
Clock for various protocols. The polarity is defined in [Advanced Decoder Settings](#).

**Chip Sel.**

Chip Select for SPI protocol. Polarity is defined in [Advanced Decoder Settings](#). If there is no Chip Select signal, clear the check box.

**Word Sel.**

Word Select for I<sup>2</sup>S protocol. Indicates which stereo channel the data belongs to.

**Display Settings**

**Name.** Type a name to display on the scope view.

**In Graph** shows the data in logic-analyzer style, on the same time axis as the analog waveform.

- Hover the mouse pointer over any decoded packet to display its contents.
- Click-and-drag the decoded data up or down the scope view.
- If the [In Graph](#) display is visible, double-click on any packet to highlight it in the table.

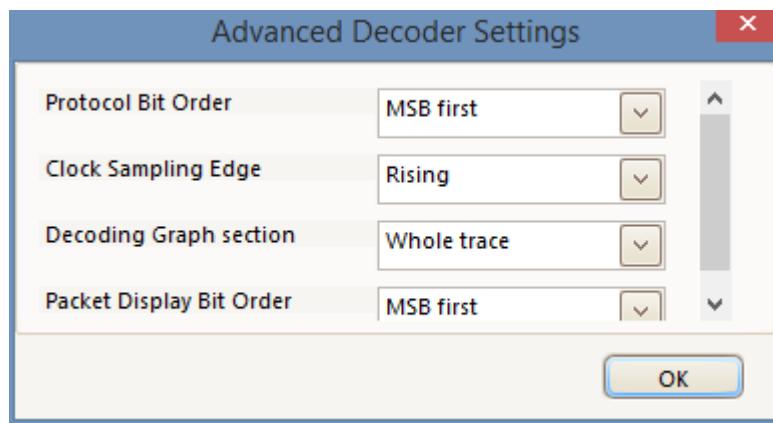
**In Table** displays the decoded data in a table in the [serial data window](#), with advanced search and filtering functions.

**Hex...** Select the format for displaying decoded data: **Hexadecimal, Binary, ASCII or Decimal.**

### Information panel

If the serial decoding parameters entered are incorrect, this panel displays an error message. If there is no error, it states the minimum number of samples required for the selected serial data format. For accurate decoding, please make sure that PicoScope is capturing at least this number of samples. The current number of samples is shown in the [Properties Window](#) ([Views menu](#) > **View Properties**).

## 6.5.4.1 Advanced Decoder Settings dialog

Location: **Tools > Serial Decoding > Advanced Settings**Purpose: Controls serial decoding settings that are not shown in the [Serial Decoding dialog](#).

Each protocol has its own selection of options from the following list:

**Protocol Bit Order**

The bit order, most-significant first or least-significant first, as defined by the protocol. For protocols with application-specific bit order, such as UART, you must set this to match to the data format.

**Decoding Graph Section**

Which section of the waveform to decode: all of it, or just the part between the [time rulers](#)

**Clock Sampling Edge**

On which edge of the clock to sample data bits

**Invert Packet Bits**

Invert all data bits

**Packet Start Edge**

For self-clocked protocols, whether rising or falling edge indicates start of packet

**Data Bits**

The number of bits in the data payload of each packet

**Parity Bits**

The number of parity-checking bits in each packet

**Stop Bits**

The number of fixed bits at the end of each packet

**ChipSelect Active State**

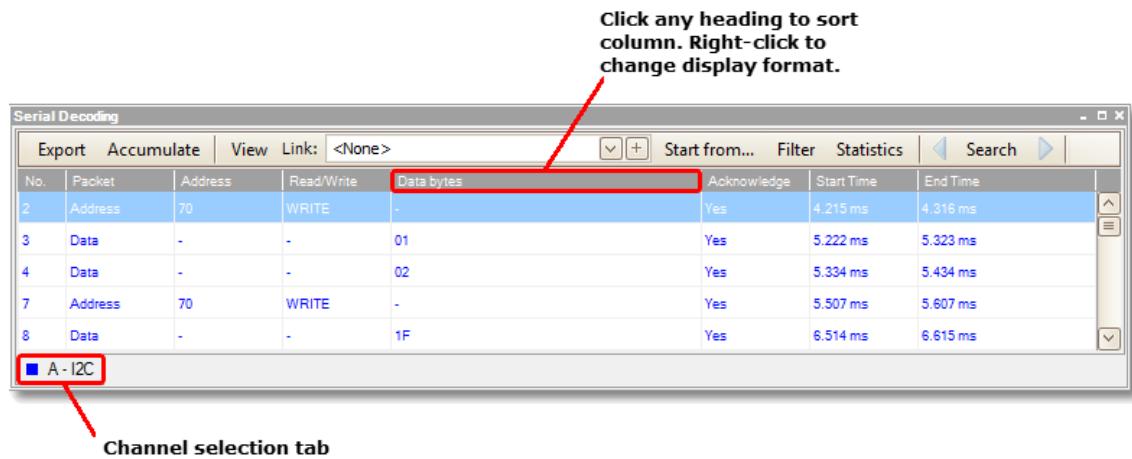
The polarity of the Chip Select signal

#### 6.5.4.2 Serial Data window

Location: [Channels toolbar](#) > **Serial Decoding button** > select **In Table** in the [Serial Decoding dialog](#)  
 Purpose: shows [decoded serial data](#) in alphanumeric format and allows advanced filtering and searching

For more information on the format of the table for each serial protocol, see [Serial protocols](#).

If you also selected **In Graph** in the [Serial Decoding dialog](#), the data will also appear in graphical form on the scope view. You can click on one of the data packets in the scope view to go to the corresponding row in the table, or you can double-click a row in the table to zoom to the corresponding frame in the scope view.



#### Control bar

The controls in the control bar are as follows:

**Export** Save the decoded data to a Microsoft Excel spreadsheet.

**Accumulate** By default, PicoScope clears the table each time the scope captures a new buffer full of data. Click this button to enter **Accumulate mode**, which continues adding serial frames to the end of the table until you click the button again.

**View** Controls which details of the data appear in the table. The sub-menus are as follows:

- Frames/Packets:** which types of frames or packets are displayed
- Fields:** which columns appear in the data table
- Display Format:** hexadecimal, binary, ASCII or decimal format
- Font Size:** the character size used in the table

**Link** This drop-down list selects a link file that maps numbers to strings. The serial data window will show these strings in place of matching numerical values.

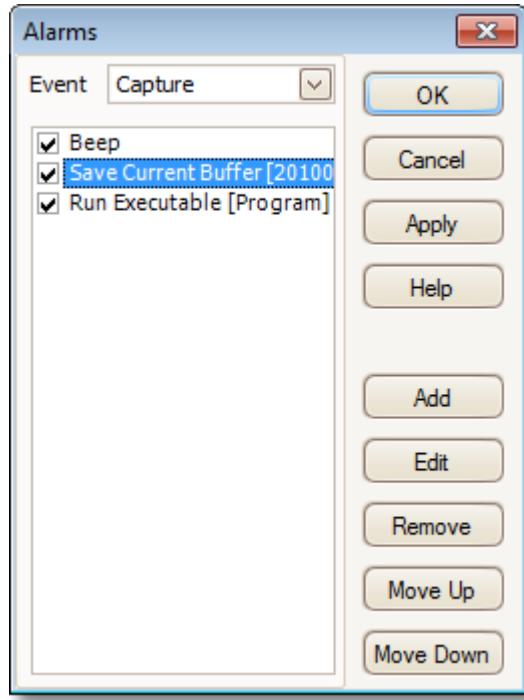
Creates a new link file. Use a text editor to add your own number-string pairs, save the file, and then open it using the drop-down control. (See also: [How to create a link file](#).)

<b>Start from...</b>	Use this button to enter a condition that PicoScope will wait for before collecting data. When PicoScope detects a packet that matches this condition, it will collect all subsequent data (subject to filtering, if used - see above) and display it in the table.
<b>Filter</b>	Click to show the filter bar, which allows you to enter arbitrary data above each column in the table. The table will display only those packets that match the data you have entered. For example, entering "6C7" in the filter box at the top of the <b>ID</b> column will cause only frames with an ID of 6C7 to be displayed.
<b>Statistics</b>	Toggle the statistics columns that list measurements such as packet start and end times, and signal voltages.
<b>Search</b>	Search for any data value in a specified column of the table.
<b>Refresh</b>	Instructs PicoScope to decode the raw data again. This is necessary if you have changed the <b>Start from...</b> condition, for example.
<b>Clear</b>	Clear all data and settings in the table. New data will appear the next time that the scope captures a waveform.

### 6.5.5 Alarms dialog

Location: **Tools > Alarms**

Purpose: gives access to the alarms feature, which specifies actions to be taken on various events



**Event:** Select the event that will trigger the alarm:

**Capture:** when a waveform is captured. If [triggering](#) is enabled, this option corresponds to a trigger event. You can therefore use this function to save a file on each trigger event.

**Buffers Full:** when the number of waveforms in the [waveform buffer](#) reaches the [maximum waveform count](#).

**Mask(s) Fail:** when any channel fails a [mask test](#).

**(Actions list):** Add an action to this list by clicking **Add**. Whenever the specified event occurs, PicoScope will execute the all the actions in the list from top to bottom.

**NOTE: For an action to execute, its check box must be set.**

**Apply:** Set up the scope according to the settings in this dialog.

**Add:** Add an event to the **Actions** list. Possible events are:

**Beep:** activate the computer's built-in sounder. 64-bit PCs redirect this sound to the headphone output.

**Play Sound:** specify the name of a [.wav](#) sound file to play.

**Stop Capture:** equivalent to pressing the red **Stop** button.

**Restart Capture:** equivalent to pressing the green **Start** button. Use only if the **Stop Capture** action was used earlier in the list.

**Run Executable:** run the specified EXE, COM or BAT program file. You can type the `%file%` variable after the program name to pass the name of the last file saved as an argument to the program. PicoScope will stop capturing while the program runs, and resume after the program terminates.

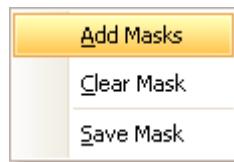
**Save Current Buffer:** save the current waveform from the buffer as a `.psdata`, `.pssettings`, `.csv` or `.mat` file. You can use the `%buffer%` variable to insert the buffer index number into the filename, or the `%time%` variable to insert the time of capture.

**Save All Buffers:** save the entire waveform buffer as a `.psdata`, `.pssettings`, `.csv` or `.mat` file.

**Trigger Signal Generator:** if the scope device has a [triggerable signal generator](#), start generating a signal.

### 6.5.6 Masks menu

Location: [Tools > Masks](#)  
 Purpose: gives control over [Mask Limit Testing](#)



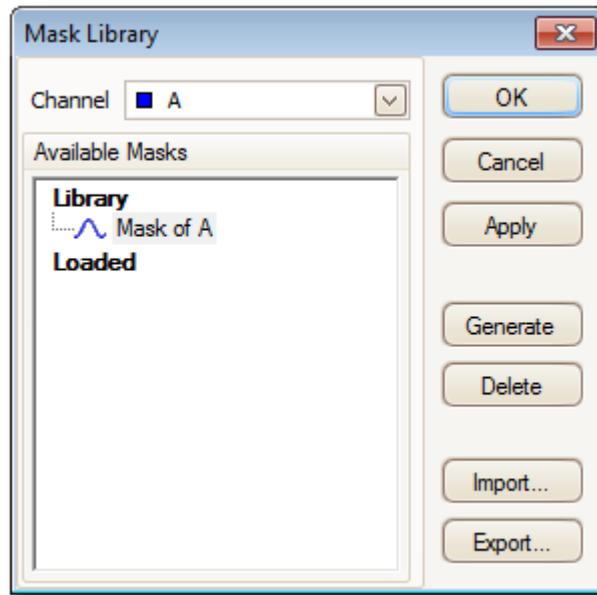
**Add Masks:** Add a mask to the display using the [Mask Library dialog](#).

**Clear Mask:** Remove the mask from the display.

**Save Mask:** Save the displayed mask to disk as a `.mask` file.

#### 6.5.6.1 Mask Library dialog

Location: [Tools > Masks](#)  
 Purpose: allows you to create, export and import masks for [Mask Limit Testing](#)



**Channel:** Select the channel to which you want to apply the mask.

**Available masks:** The **Library** section shows all the masks that you have saved in the past and not deleted. The **Loaded** section shows all the masks currently in use.

**Generate:** Create a new mask based on the last waveform captured from the selected channel. Opens the [Generate Mask dialog](#).

**Import:** Load a mask that has previously been saved as a `.mask` file.

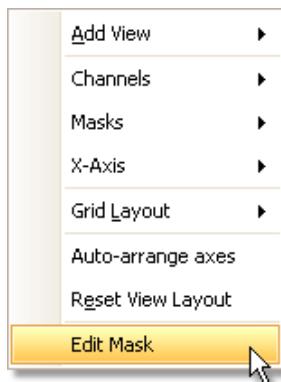
**Export:** Save a mask as a `.mask` file for future import.

**Apply:** Use the selected mask on the selected channel but remain in the **Mask Library dialog**.

**OK:** Use the selected mask on the selected channel and return to the [scope view](#).

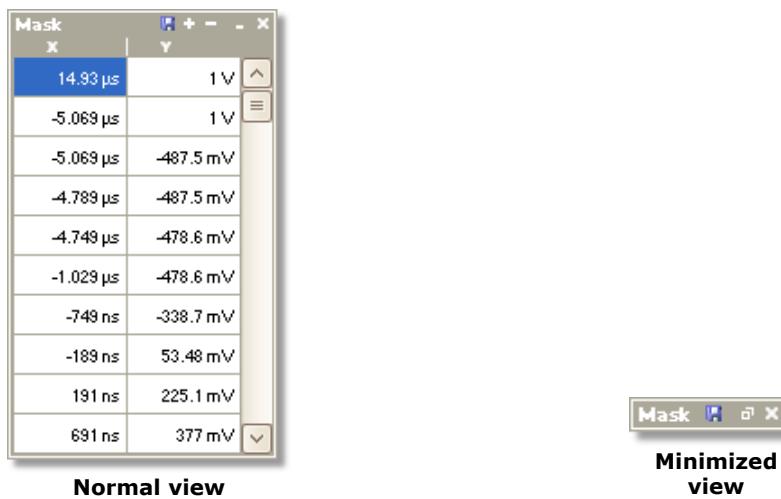
### 6.5.6.2 Editing a mask

To edit a mask in **Mask Limit Testing** mode, right-click on the [scope view](#) and select **Edit Mask**:



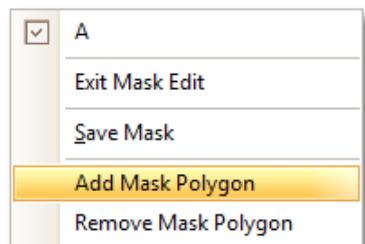
A mask is made up of one or more shapes called **polygons**. Click on the polygon you wish to edit. PicoScope will then draw edit handles on the selected mask polygon and display the mask edit box. If you drag any of the handles to edit the polygon, the statistical results will be updated immediately.

The mask edit box looks like this:



If the edit box is not immediately visible, it may have been minimized; in which case, click the restore button: . If you edit the coordinates of a vertex, the statistical results will be updated immediately. You can also export the mask to a `.mask` file with the export button: . Use the + and - buttons to add or remove vertices. The minimize button has its usual function. To leave mask editing mode, close the mask edit box with the close (X) button.

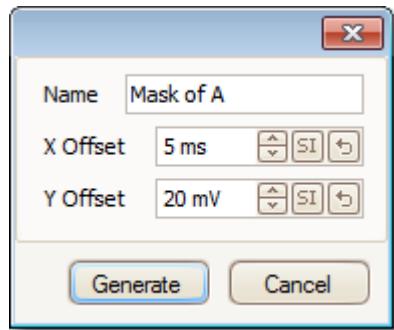
To add or remove an entire polygon, right-click the scope view and select either the **Add Mask Polygon** or the **Remove Mask Polygon** command:



## 6.5.6.3 Generate Mask Dialog

Location: [Mask Library dialog > Generate](#)

Purpose: allows you to set parameters for the automatically generated mask. PicoScope will then create a new mask based on the last captured waveform.



**Name:** PicoScope automatically chooses a name for the new mask. You can edit the name in this box.

**X Offset:** The horizontal distance between the waveform and the mask.

/ This button toggles the offset value between absolute units (SI) and relative units (% of full scale).

This button resets the offset value to its default.

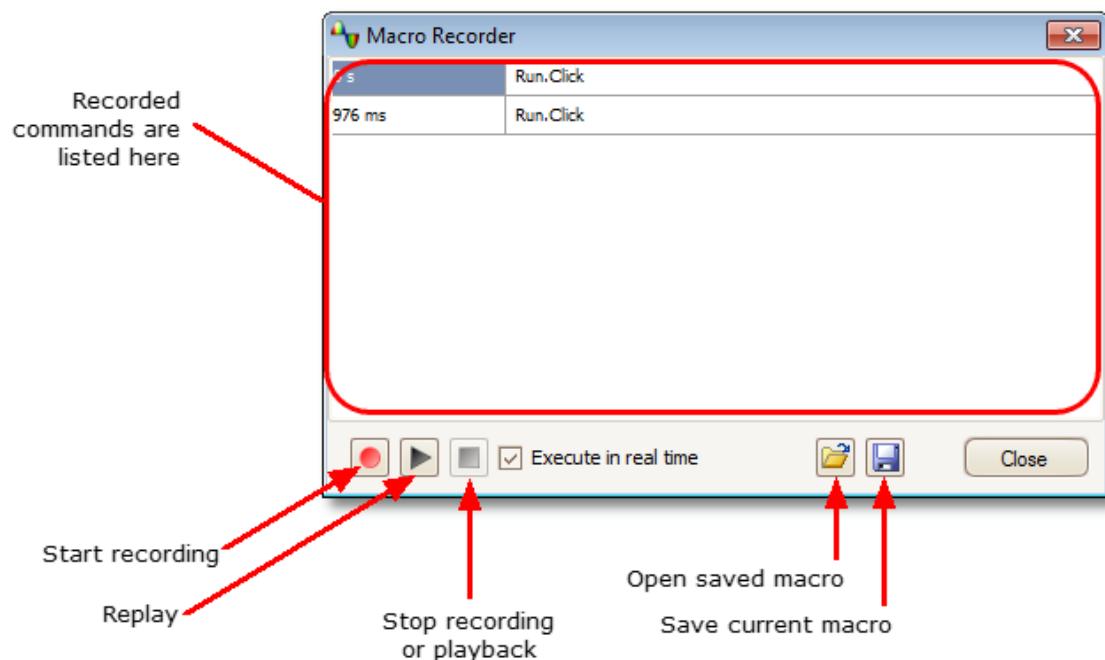
**Y Offset:** The vertical distance between the waveform and the mask.

### 6.5.7 Macro Recorder

Location: **Tools > Macro Recorder**

Purpose: records a sequence of commands to be replayed later

The **Macro Recorder** helps when you wish to execute a series of commands repeatedly. It saves all commands to a `.psmacro` file, which can be modified using an XML editor.



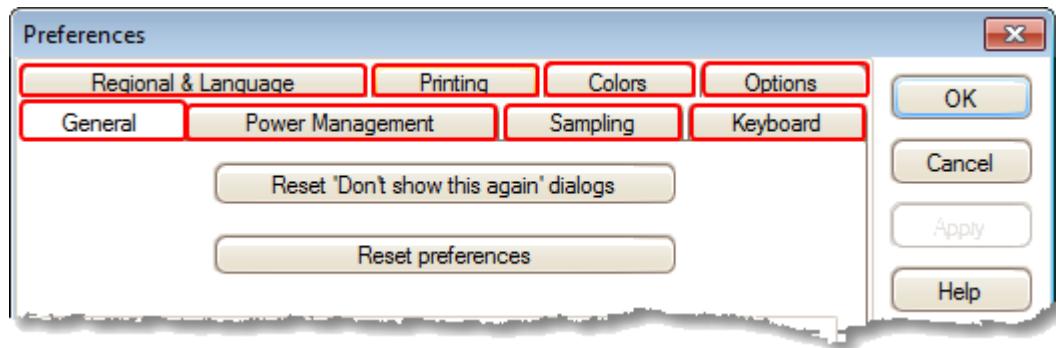
**Execute in real time:** Play back the macro at the same speed as when recorded. Without this option, playback will be as fast as possible.

Note: `.psmacro` files can also be played back from the [PicoScope command line](#).

### 6.5.8 Preferences dialog

Location: **Tools > Preferences**

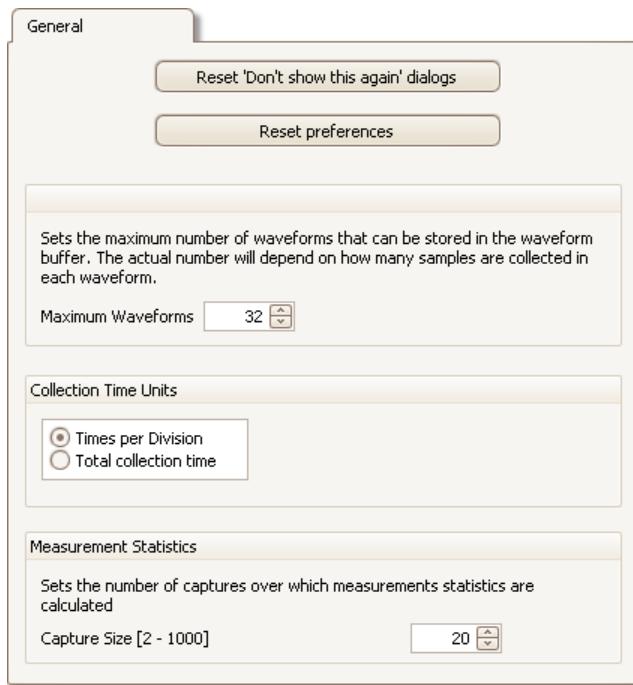
Purpose: Allows you to set options for the PicoScope software. Click one of the tabs in the picture below to learn more.



## 6.5.8.1 General page

Location: **Tools > Preferences > General**

Purpose: contains general controls for PicoScope

**Reset 'Don't show this again' dialogs**

Restore any missing dialogs that you asked PicoScope not to show again.

**Reset preferences**

Set all preferences back to their default values.

**Waveform Buffer**

**Maximum Waveforms:** This is the maximum number of waveforms that PicoScope will store in the [waveform buffer](#). You can select a number from 1 to the maximum allowed by the oscilloscope that is connected: see the scope specifications for details). The actual number of waveforms stored depends on the available memory and the number of samples in each waveform.

**Collection Time Units**Change the mode of the **Timebase** control in the [Capture Setup toolbar](#).

**Times per division:** the **Timebase** control displays time units per division - for example, '5 ns /div'. Most laboratory oscilloscopes display timebase settings in this way.

**Total collection time:** the **Timebase** control displays time units for the entire width of the scope view - for example, '50 ns'.

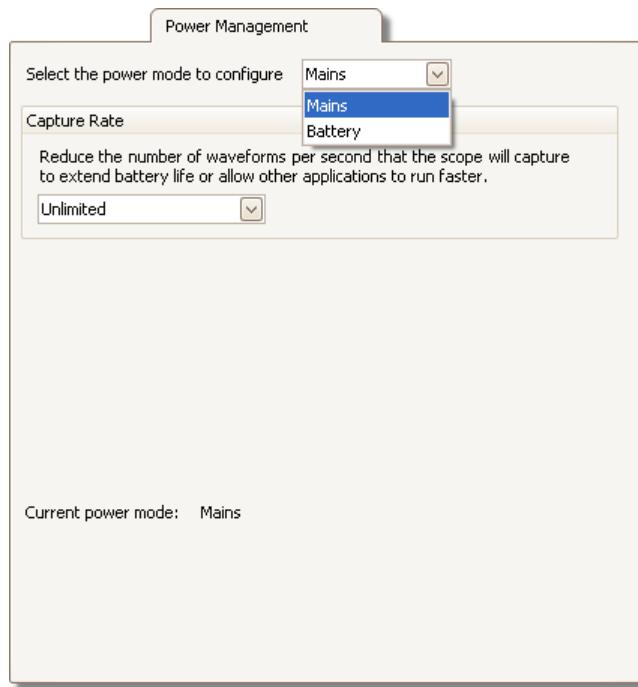
**Measurement Statistics**

**Capture Size** - the number of successive captures that PicoScope uses to calculate the statistics in the [Measurements Table](#). A larger number produces more accurate statistics but causes them to be updated less frequently.

### 6.5.8.2 Power management page

Location: **Tools > Preferences > Power Management**

Purpose: controls features of the oscilloscope that affect its power consumption



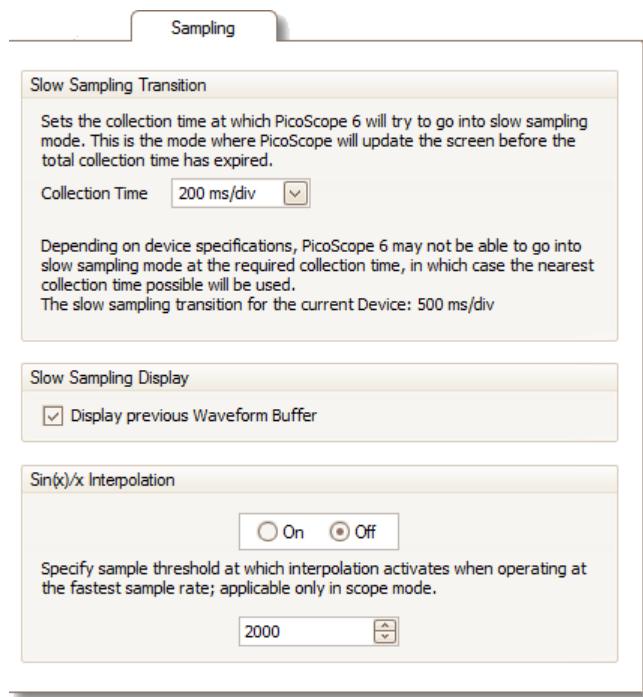
#### Capture Rate

This control limits the speed at which PicoScope captures data from the scope device. The other PicoScope settings, the type of [scope device](#) and the speed of the computer will all affect whether this limit can actually be reached. PicoScope automatically selects the appropriate limit according to whether your computer is running on batteries or on mains (line) power.

The settings are in captures per second. By default, the capture rate is set to "Unlimited" when your computer is running on **Mains** (line) power, for maximum performance. If other applications run too slowly on your PC whilst PicoScope is capturing, then reduce the capture rate limit. When your computer is running on **Battery** power, PicoScope imposes a performance limit to save the battery. You can increase this limit manually, but this will cause the battery power to drain very quickly.

## 6.5.8.3 Sampling page

Location: **Tools > Preferences > Sampling**  
 Purpose: controls the sampling behavior of the oscilloscope



### Slow Sampling Transition

In normal (fast) sampling mode, PicoScope collects enough data to fill the screen and then redraws the whole view at once. This method is suitable for fast timebases, when the screen is redrawn many times each second, but with slow timebases it can cause an unacceptable delay before the data appears on the screen. To avoid this delay, PicoScope automatically switches to slow sampling mode, in which the scope trace advances gradually across the screen as the scope captures data.

The **Collection Time** control lets you select the timebase at which PicoScope switches to slow sampling mode.

### Slow Sampling Display

When this box is checked, PicoScope displays the previous waveform in the buffer while gradually redrawing the new waveform over the top of it. Thus, at any time, the left-hand side of the view shows the beginning of the new waveform, while the right-hand side shows the end of the previous waveform. A vertical bar separates the two waveforms.

### Sin(x)/x Interpolation

When the number of pixels across the scope view is greater than the number of samples in the waveform buffer, PicoScope interpolates - that is, it fills the space between the samples with estimated data. It can either draw straight lines between the samples (linear interpolation) or connect them with smooth curves ( $\sin(x)/x$  interpolation). Linear interpolation makes it easier to see where the samples are, which is useful for high-accuracy measurements, but results in a jagged waveform.  $\sin(x)/x$  interpolation gives a smoother waveform but disguises the true locations of the samples, so should be used with care when the number of samples on the screen is low.

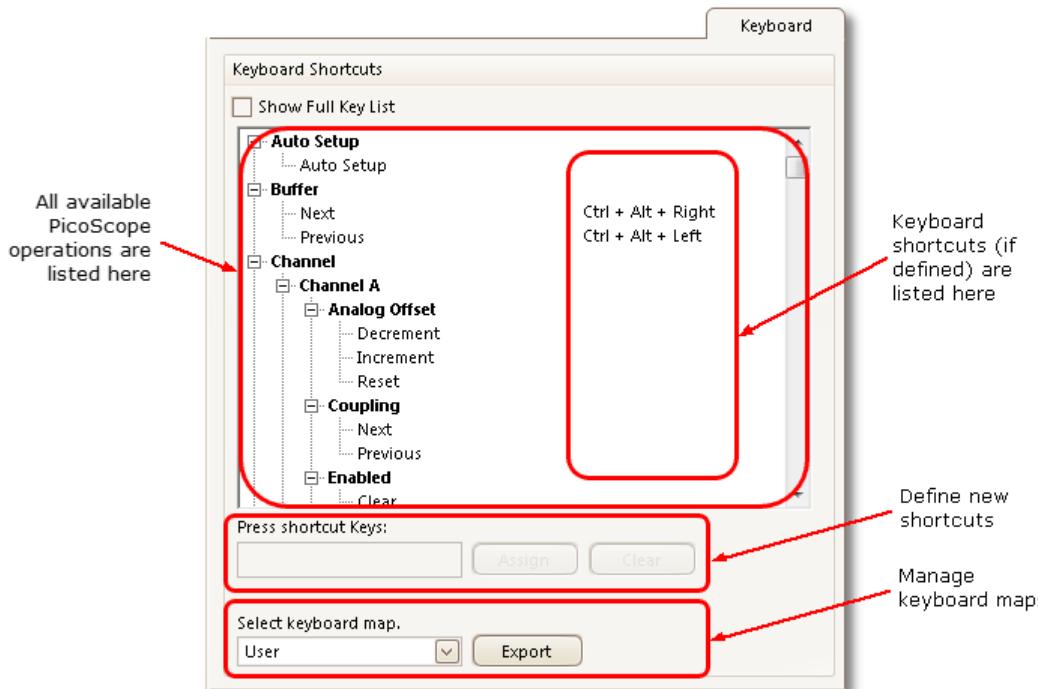
You can adjust the number of samples below which  $\sin(x)/x$  interpolation is switched on.  $\sin(x)/x$  interpolation is used only on the scope's fastest timebase.

## 6.5.8.4 Keyboard page

Location: **Tools > Preferences > Keyboard**

Purpose: displays, and allows you to edit, keyboard shortcuts

A keyboard shortcut is a combination of keys that can be pressed on the keyboard to activate a PicoScope operation.

**Keyboard Shortcuts**

This is a list of PicoScope operations and their associated keyboard shortcuts (if defined). The extent of the list depends on the [Show Full Key List](#) option (see below).

To edit or add a keyboard shortcut:

- Scroll the list of PicoScope commands until the required operation is visible.
- Select the required operation.
- Select the 'Press shortcut keys:' box.
- Press the required key combination on the keyboard.
- Click **Assign**.

**Show Full Key List**

Check this box to show all available operations. By default, only the most common operations are listed, plus any other operations that have a keyboard shortcut assigned to them.

**Keyboard Maps**

A set of keyboard shortcuts is called a **map**. You can define multiple maps for different applications.

**Default:** This map cannot be edited. Use it to return to the factory-defined basic shortcuts.

**Advanced:** This is another factory-defined map that cannot be edited. It contains a more comprehensive set of shortcuts.

**User:** This is the map that you most recently created or imported. It is preserved between PicoScope sessions.

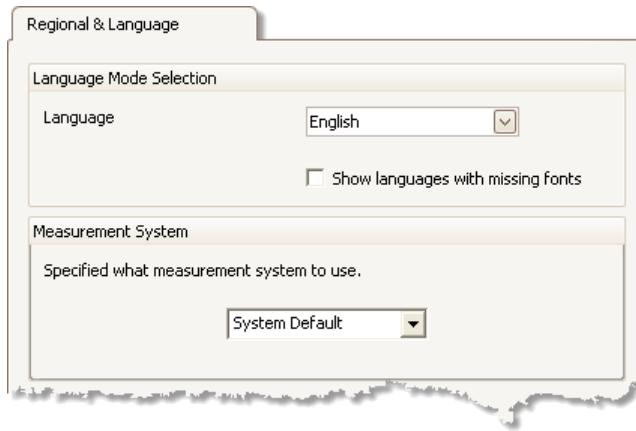
**Import:** Load a keyboard map from a `.pskeys` file.

**Export:** Save the current keyboard map to a `.pskeys` file.

## 6.5.8.5 Regional &amp; Language page

Location: **Tools > Preferences > Regional & Language**

Purpose: lets you select the language and other location-dependent settings for PicoScope's user interface

**Language**

Select, from the drop-down box, the language you wish to use for the PicoScope 6 user interface. PicoScope will ask you to restart the program before switching to the new language.

**Measurement System**

Select metric or U.S. units.

## 6.5.8.6 Printing page

Location: **Tools > Preferences > Printing**

Purpose: lets you enter the details that will appear at the bottom of printed output

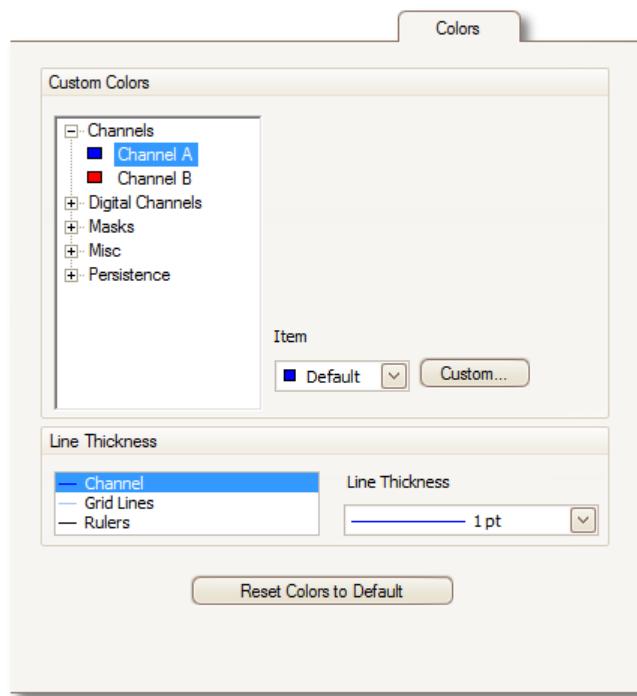
**Default Print Settings**

When you print a view from the [File menu](#), these details will be added to the bottom of the page.

## 6.5.8.7 Colors page

Location: **Tools > Preferences > Colors**

Purpose: lets you set the colors for various parts of the user interface

**Custom Colors**

These controls let you specify the colors for various parts of the PicoScope screen:

<b>Channels</b>	the trace color for each <a href="#">scope channel</a>
<b>Digital Channels</b>	if you have a <a href="#">mixed-signal oscilloscope (MSO)</a> , the color of each channel can be set here
<b>Masks</b>	the mask areas in <a href="#">Mask Limit Testing</a>
<b>Misc</b>	miscellaneous items:
<b>Grid lines</b>	the horizontal and vertical lines on the <a href="#">graticule</a>
<b>Background</b>	the area behind the waveforms and graticule. (In <a href="#">persistence mode</a> this setting can be overridden by the <a href="#">Persistence Options dialog</a> .)
<b>Live trigger</b>	the <a href="#">trigger marker</a> for the current trigger position
<b>Trigger</b>	secondary trigger marker (appears when the live trigger has moved since the last waveform capture)
<b>Horizontal axis</b>	the numbers across the bottom of each <a href="#">view</a> , which usually indicate time measurements
<b>Rulers</b>	the horizontal and vertical <a href="#">rulers</a> that you can drag into position to help measure features on the waveform

**Persistence**

the three colors to use for each channel in digital color [persistence mode](#). The top color is used for the most frequently hit pixels, the middle and bottom colors for the less and least frequently hit pixels.

**Line Thickness**

These controls let you specify the thickness of the lines drawn on the [scope](#) and [spectrum](#) views:

**Channel**

the waveforms and spectrum traces for all scope channels

**Grid Lines**

the horizontal and vertical lines on the [graticule](#)

**Markers**

the horizontal and vertical [rulers](#) that you can drag into position to help measure features on the waveform

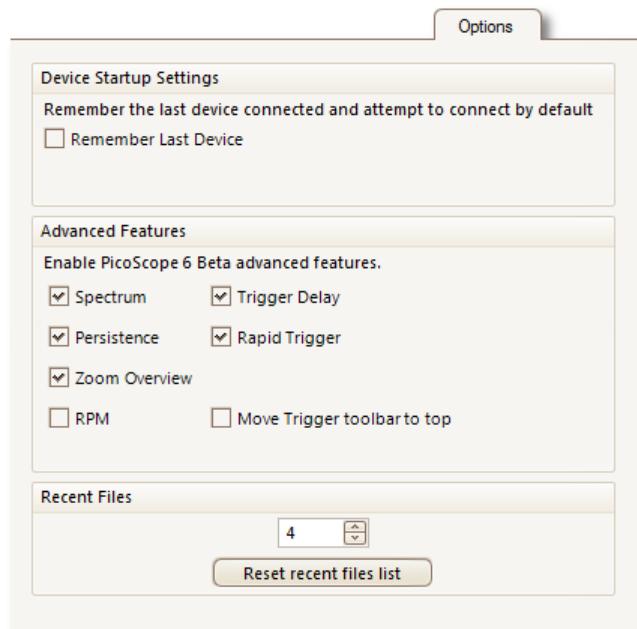
**Reset Colors to Default**

Resets all of the color and line thickness settings to their default values.

## 6.5.8.8 Options page

Location: **Tools > Preferences > Options**

Purpose: lets you set various options that control the way that PicoScope 6 works

**Device Startup Settings**

**Remember Last Device.** This option is used when PicoScope finds more than one scope device. If the check box is ticked, PicoScope will attempt to use the same device that was used last time. Otherwise it will use the first device available.

**Advanced Features**

The advanced [capture modes](#) are enabled by default in PicoScope 6 and disabled by default in PicoScope 6 Automotive. Regardless of the version that you have, you can enable or disable these features using the following options:

**Spectrum Persistence**

[Spectrum view](#) and [spectrum analyzer](#) features Digital Color, Analog Intensity and custom [persistence display](#) modes

**Zoom Overview**

A window that appears when you [zoom in](#), to help to move around large waveforms with the minimum of mouse clicks

**RPM**

Revolutions per minute, displayed alongside hertz in the [frequency legend](#)

**Trigger Delay  
Rapid Trigger**

The time-delay control in the [Triggering toolbar](#). The 'Rapid' entry in the Trigger mode control in the [Triggering toolbar](#).

**Move Trigger toolbar to top**

The [toolbar](#) containing the [Start/Stop](#), [Triggering](#), [Measurements](#) and [Rulers](#) controls is at the bottom of the PicoScope window by default. This option moves it to the top.

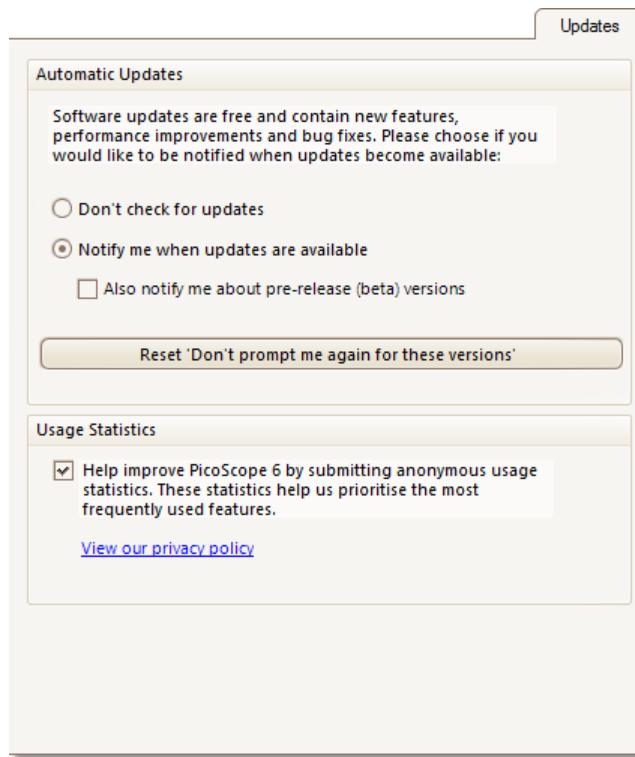
**Recent Files**

The maximum number of files listed in the [File > Recent Files](#) menu. Click the button to clear the list.

## 6.5.8.9 Updates page

Location: **Tools > Preferences > Updates**

Purpose: configures the automatic update checker and related services

**Automatic Updates**

**Don't check for updates.** If you prefer, you can visit [www.picotech.com](http://www.picotech.com) periodically to check for updates.

**Notify me...** PicoScope will regularly check for updates to your software. (Requires an internet connection.)

**Reset...** If you checked the "Don't remind me again..." option in the Software Updates dialog, this button enables reminders again.

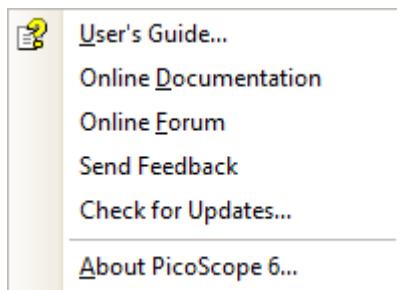
**Usage Statistics**

To help us to continue improving PicoScope, the program periodically sends back to our developers anonymous information about which features you use the most. This information does not contain your name, email address or any other personal details, but does allow us to identify which country you are in. If you do not wish to send us this information, uncheck this box.

## 6.6 Help menu

Location: **Help**

Purpose: gives access to the PicoScope 6 User's Guide and related information



**User's Guide** This is the most complete source of information on PicoScope.

**Online Documentation** Find manuals and training guides for Pico Technology products.

**Online Forum** Request technical support and discuss questions with fellow PicoScope users. New software features are sometimes announced here before they make it into the **User's Guide**.

**Send Feedback** Send us suggestions for improving PicoScope.

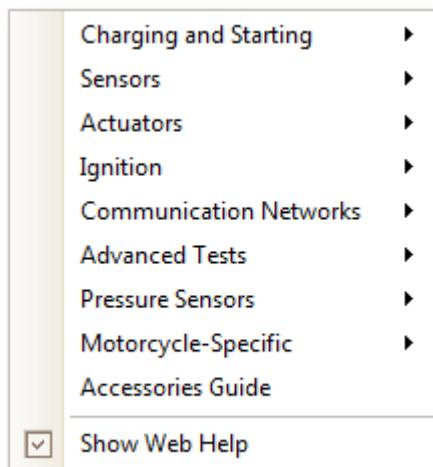
**Check for Updates** Check online for later versions of PicoScope. If you have both stable and beta versions installed, this checks for updates to both of them. You can configure the settings for automatic update checking under the **Tools > Preferences > Updates** menu.

**About PicoScope 6** Displays useful information such as the model and serial numbers of your scope, and software and driver version numbers.

## 6.7 Automotive menu (PicoScope Automotive only)

Location: **[Menu bar](#) > [Automotive](#)**

Purpose: gives access to a database of preset tests



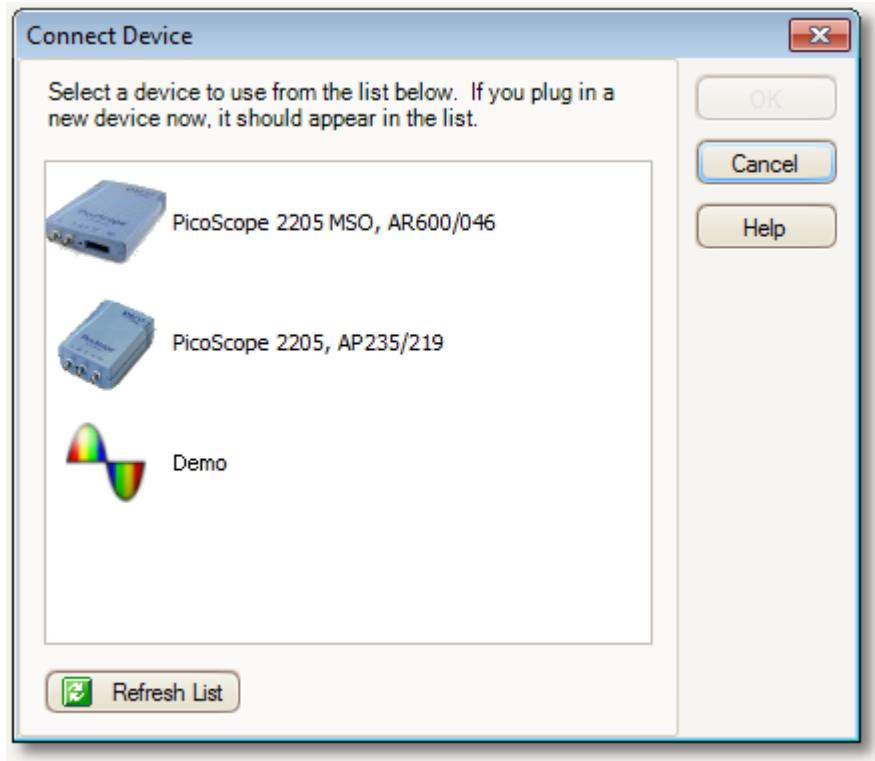
**Show Web Help:** When this box is checked, selecting a preset test also opens a web page with instructions and technical information. Clear this box to disable web help.

1. Select a preset test.
2. PicoScope opens the relevant information page that explains how to setup the test for the scope that is connected, run the test and interpret the results (a few tests do not have an information page).
3. PicoScope displays an example waveform.
4. PicoScope configures itself with the necessary settings. In most cases all you need to do is press the space bar to start the test.

## 6.8 Connect Device dialog

Location: **File > Connect Device**  
or plug in a new device

Purpose: when PicoScope finds more than one available [scope device](#), this dialog allows you to select which one to use



See "[How to change to a different device](#)" if you wish to switch to a different scope device later.

### Procedure

- ➊ Wait for a list of devices to appear. This may take a few seconds.
- ➋ Select a device and click **OK**.
- ➌ PicoScope will open a [scope view](#) for the selected scope device.
- ➍ Use the [toolbars](#) to set up the device and the [scope view](#) to display your signals.

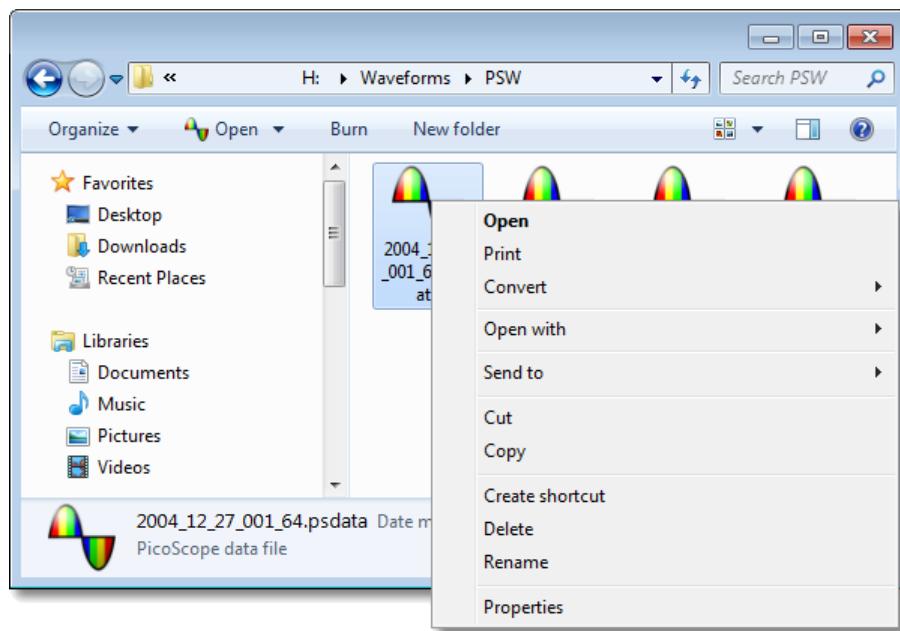
### Demonstration mode

If you start PicoScope with no device connected, the **Connect Device dialog** automatically appears with a '**Demo**' (demonstration) device as one of the options. This is a virtual device that you can use to experiment with the features of PicoScope. If you select the **Demo** device and click **OK**, PicoScope adds a [Demo Signal Generator button](#) to the toolbar. Use this button to set up the test signals from your **Demo** device.

## 6.9 Converting files in Windows Explorer

You can convert PicoScope data files to other formats for use in other applications, or to different forms of data for use with PicoScope.

The easiest way to do this conversion is through the context menu in **Windows Explorer**. The context menu is the menu that pops up when you right-click it with the mouse or activate it with the "menu" button on a Windows keyboard. When you install PicoScope, a "**Convert**" entry is added to the context menu to enable you to convert PicoScope data files.



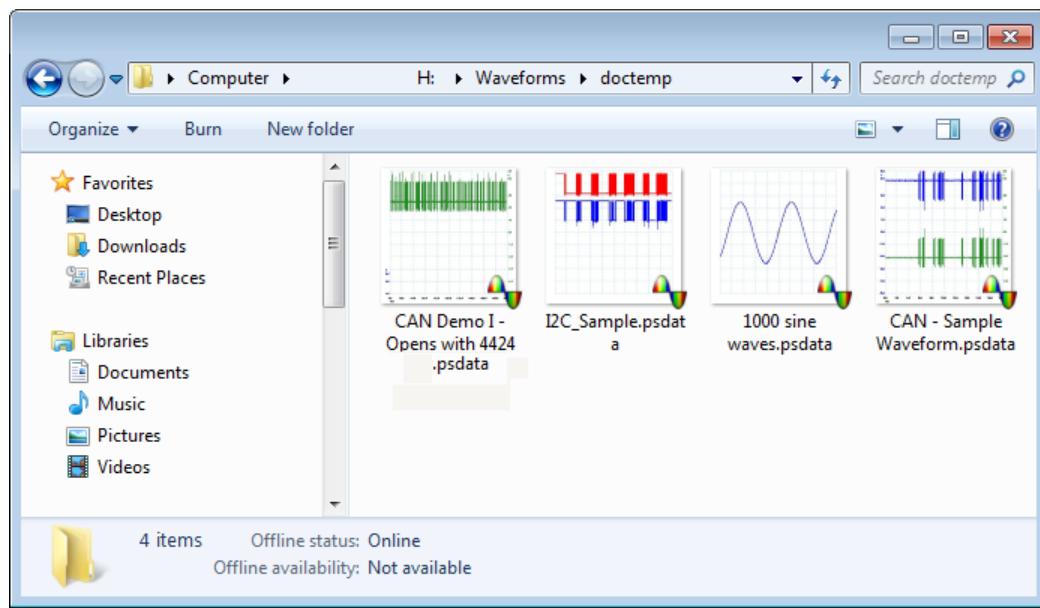
The PicoScope context menu in Windows Explorer

### Converting to PicoScope 6.2.4 format

The example above shows four pre-existing PicoScope data files represented by standard PicoScope icons. PicoScope 6.2.4 introduced a new feature that allows PicoScope data files to appear as waveforms instead of icons. To enable this feature for old data files, you need to convert them to the new format using the context menu of Windows Explorer.

- ➊ If PicoScope is running, close it.
- ➋ In Windows Explorer, right-click a PicoScope data file.
- ➌ Select **Convert > All waveforms > .psdata**. A PicoScope icon  will appear in the Windows notification area while the conversion is in progress.
- ➍ PicoScope will ask you to confirm that you wish to overwrite the .psdata file with a new version. Click **Yes**.
- ➎ Wait for Windows Explorer to update the display.
- ➏ Repeat for all .psdata files.

The `.psdata` files should now appear as in this picture:



### Converting to other formats

For all of these conversions, you can choose either "**All waveforms**" or "**Current waveform**". A `.psdata` file can contain either a single waveform or the entire contents of the waveform buffer, which can hold a number of waveforms from successive trigger events. If the `.psdata` file contains more than one waveform, then you can choose to convert all of them or just the one that was last viewed in PicoScope.

- Right-click a PicoScope data file.
- To convert all waveforms in the file, select **Convert > All waveforms** or **Convert > Current waveform** and then the file format that you require. A PicoScope icon  will appear in the Windows notification area while the conversion is in progress.

### Complex operations

For more complex operations, such as converting all files in a directory, you can run PicoScope in a command window (see [Command-line syntax](#)).

## 7 Toolbars and buttons

A **toolbar** is a collection of buttons and controls with related functions.

## 7.1 Advanced Options toolbar

The **Advanced Options toolbar** controls **Phase (or Rotation) Rulers**, **Notes** and (in PicoScope Automotive only) **Channel Labels**.



It contains the following buttons:

**Rulers** Opens the [Ruler Settings dialog](#) that controls the **Phase rulers** (or in PicoScope Automotive, **Rotation rulers**)

**Notes** Displays the [Notes](#) at the bottom of the window

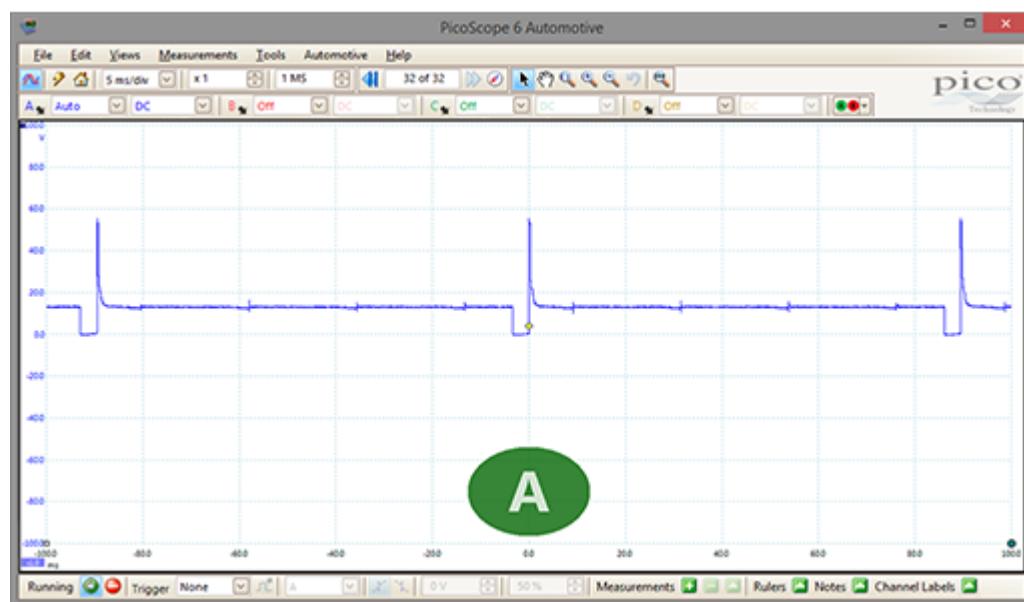
**Channel Labels** (PicoScope Automotive only) Displays the [Channel Labels](#) at the bottom of the window

This toolbar normally resides at the bottom of the program window, but can be moved to the top using the [Tools > Preferences > Options > Bottom toolbar at top](#) control.

## 7.2 ConnectDetect

A feature included on our PicoScope 4225 and 4425 PicoScopes is **ConnectDetect**. This feature is only available when DC coupling is selected, and indicates whether a channel probe has a good physical connection to the signal under test or not. With **ConnectDetect** active, the Channel LED for an enabled channel will either be green to indicate that the test probe is directly connected to a component, or red to indicate that it is not. This will also be shown on the PicoScope screen, and can be seen below (where Channel A is active).

To enable **ConnectDetect**, click the **ConnectDetect** button (  ) in the software.



## 7.3 Channels toolbar

The **Channels toolbar** controls the settings for each vertical input [channel](#). The screen shot below shows the toolbar for a two-channel [scope device](#), but different scope devices may have different numbers of channels. (See also [PicoLog 1216 toolbar](#) that is used for the PicoLog 1000 Series.)



Each channel has its own set of buttons:



**Channel Options button.** Opens the [Channel Options menu](#) with options for [probes](#), [resolution enhancement](#), [scaling](#) and [filtering](#).



**Range control.** Sets up the scope device to capture signals over the specified range of values. The list of options depends on the selected [scope device](#) and [probe](#). A red warning symbol - ! - appears if the input signal exceeds the selected range. If you select **Auto**, PicoScope will continually adjust the vertical scale so that the height of the waveform fills as much of the view as possible.



**Coupling Control.** Sets up the input circuitry.

**AC coupling:** rejects frequencies below about 1 Hz.

**DC coupling:** accepts all frequencies from DC to the scope's maximum bandwidth.

**50Ω DC:** low-impedance option (see [Device feature table](#)).

**Accelerometer:** switches on the current source output for [IEPE](#)-enabled scopes such as the PicoScope 4224 IEPE. The User's Guide for the oscilloscope has details of the [IEPE](#) channel specifications.

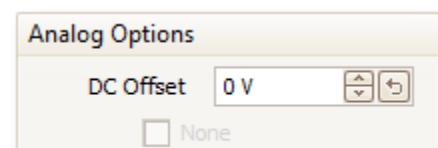
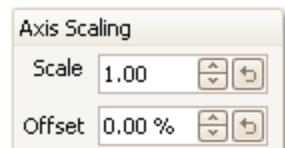
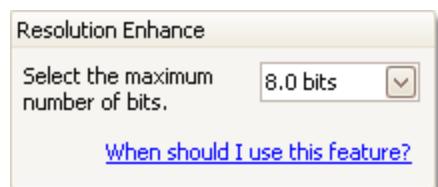
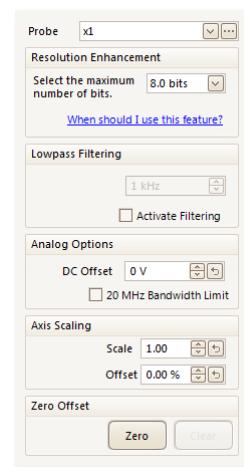
**Frequency:** enable the built-in frequency counter if available. Only one channel at a time can be operated in this mode. For availability, see [Device feature table](#).



[Digital Inputs button](#) ([MSOs](#) only).

### 7.3.1 Channel Options menu

The **Channel Options menu** appears when you click the **Channel Options button** (for example:  ) on the [Channels toolbar](#).



**Probe list.** Indicates the probe currently in use and allows you to select a different one. Use it to tell PicoScope what type of probe is connected to a channel. By default, the probe is assumed to be **x1**, which means that a one-volt signal at the input to the probe will appear as one volt on the display.

**Expand probe list.** Click this to select from a list of probes.

**Open Custom Probes dialog.** The [Custom Probes dialog](#) allows you to edit your library of custom probes.

**Resolution enhancement.** Allows you to increase the effective resolution of your scope device using [Resolution enhancement](#). The number in this box is a target value that the software will attempt to use whenever possible.

**Axis Scaling.** These are the [axis scaling controls](#) that allow you to set the scale and offset for each vertical axis individually.

**Analog Options.** Options that can be applied to the oscilloscope input hardware, if the oscilloscope hardware supports them.

**DC Offset:** an offset voltage added to the analog input before digitization. For availability, see the [Device feature table](#).

**Bandwidth Limit:** a fixed-frequency single-pole analog filter. This can be useful for rejecting noise and harmonics that would otherwise cause aliasing. For availability, see the [Device feature table](#).



**Lowpass Filtering.** An independent digital [lowpass filter](#) for each input channel, with programmable cut-off frequency. This can be useful for removing noise from your signal to make more accurate measurements. For availability, see the [Device feature table](#).



**Zero offset.** Digitally removes any offset from the input channel. Before starting this operation, remove any input signal from the selected channel and short-circuit the input. Click **Zero** to begin the adjustment. Click **Clear** to restore the input to its uncorrected state.

#### 7.3.1.1 Resolution enhancement

**Resolution enhancement** is a technique for increasing the effective vertical resolution of the scope at the expense of high-frequency detail. In some scope operating modes PicoScope may reduce the number of samples available to maintain display performance.

For this technique to work, the signal must contain a very small amount of Gaussian noise, but for many practical applications this is generally supplied by the scope itself and the noise inherent in normal signals.

The resolution enhancement feature uses a flat moving-average filter. This acts as a low-pass filter with good step response characteristics and a very slow roll-off from the pass-band to the stop-band.

Some side-effects will be observed when using resolution enhancement. These are normal and can be counteracted by reducing the amount of enhancement used, increasing the number of samples captured or changing the timebase. Trial and error is usually the best way to find the optimum resolution enhancement for your application. The side-effects include:

- Widened and flattened impulses (spikes)
- Vertical edges (such as those of square waves) turned into straight-line slopes
- Inversion of the signal (sometimes making it look as if the trigger point is on the wrong edge)
- A flat line (when there are not enough samples in the waveform)

#### Procedure

- Click the **Channel Options** button  in the [Channel Setup toolbar](#).
- Use the **Resolution Enhancement** control in the [Advanced Options menu](#) to select the effective number of bits, which can be equal to or greater than the [vertical resolution](#) of your scope device.

## Quantifying Resolution Enhancement

The table below shows the size of the moving-average filter for each resolution enhancement setting. A bigger filter size requires a higher sampling rate to represent a given signal without significant side-effects (as detailed above).

Resolution enhancement $e$ (bits)	Number of values $n$
0.5	2
1.0	4
1.5	8
2.0	16
2.5	32
3.0	64
3.5	128
4.0	256

**Example.** Your scope device is a PicoScope 5204 (resolution = 8 bits). You have selected an effective resolution of 9.5 bits. The resolution enhancement is therefore:

$$e = 9.5 - 8.0 = 1.5 \text{ bits.}$$

The table shows that this is achieved using a moving average of:

$$n = 8 \text{ samples.}$$

This number gives a guide to what sort of filtering effect the resolution enhancement will have on the signal. The best way of seeing the actual low-pass filter effect is to add a spectrum view and look at the shape of the noise floor (try dragging the y-axis upwards to see the noise more clearly).

## Related topics

See [Hardware resolution](#) (applies to Flexible Resolution Oscilloscopes only).

## 7.3.1.2 Axis scaling controls

The **axis scaling controls** are control boxes that let you change the scale and offset of each vertical axis individually. If the axis belongs to a [reference waveform](#) then you can also adjust its delay relative to the live waveforms.



There are two ways to open the axis scaling control: -

- For any channel displayed in a [view](#): click the colored scaling button ( at the bottom of the vertical axis
- For any input channel: Click the [Channel Options button](#) in the [Channels toolbar](#)



**Scale control.** Increase to magnify the waveform, decrease to reduce it. The vertical axis rescales accordingly so that you can always read the correct voltage from the axis. Click the reset button ( to return to a scale of 1.0. The scaling button always shows the selected scale.



**Offset control.** Increase to move the waveform up the display, decrease to move it down. The vertical axis shifts accordingly so that you can always read the correct voltage from the axis. Adjusting this control is equivalent to clicking and dragging the vertical axis. Click the reset button ( to return to an offset of 0.00%.



**Delay control (for reference waveforms only).** Increase to move the waveform to the left relative to the timing reference point, decrease to move it to the right. Click the reset button ( to return to a delay of 0 s.

The location of the timing reference point depends on which [trigger mode](#) PicoScope is in. If the trigger mode is **None** then the delay is measured relative to the left-hand edge of the display. In all other trigger modes, the delay is measured relative to the [trigger](#) marker.



**Send to back.** Draws the channel behind all others. Use if the channel is obscuring another channel of interest.



**Bring to front.** Draws the channel in front of all others. Use if the channel is hidden behind another.

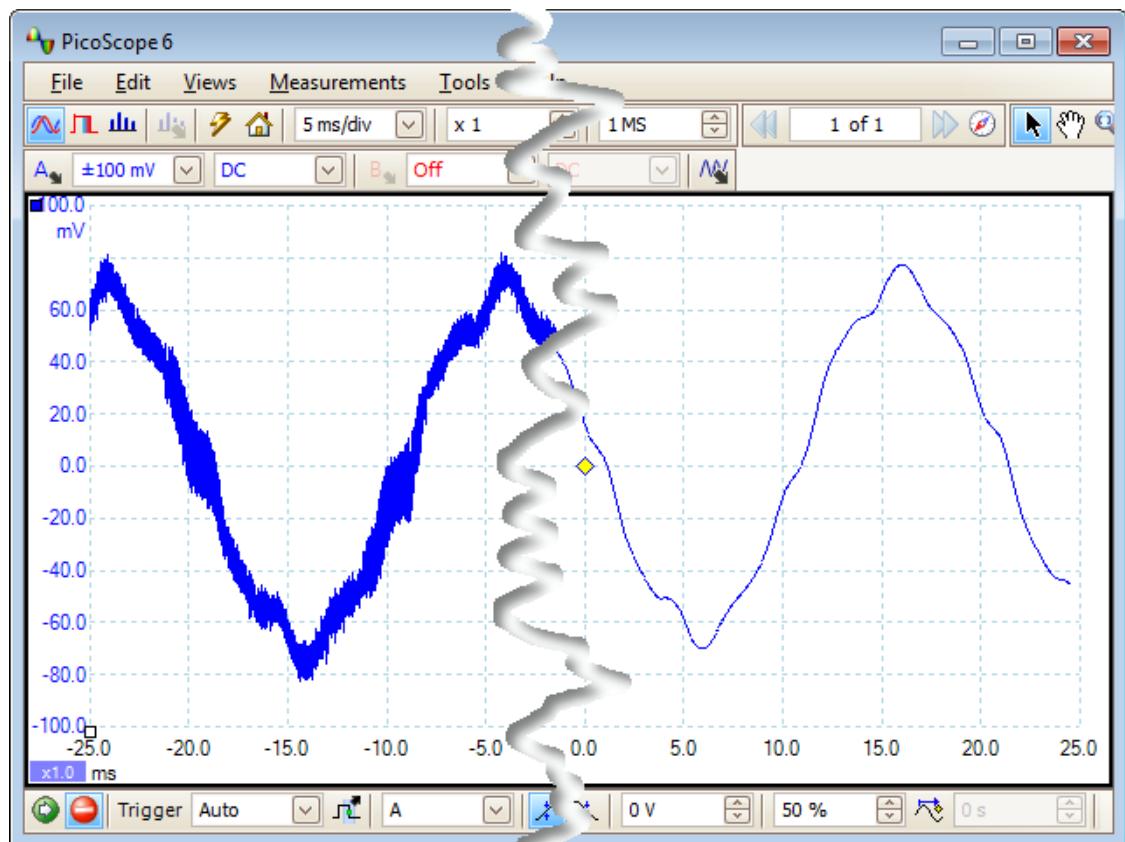
### 7.3.1.3 Lowpass filtering

The **lowpass filtering** feature can reject high frequencies from any selected input channel. The filtering control is found in the [Advanced Channel Options dialog](#), which is opened by clicking the **Channel Options button** (A) for the relevant channel on the [Channels toolbar](#). The control determines the cut-off frequency of the filter, which must be below half the sampling rate shown in the [Properties sheet](#).



For availability, see the [Device feature table](#).

Lowpass filtering is useful for rejecting noise. The split screenshot below shows the effect of applying a 1 kHz lowpass filter on a noisy signal. The underlying shape of the signal is preserved but the high-frequency noise is eliminated:



Left: before lowpass filtering. Right: after 1 kHz lowpass filtering.

### Filter details

The lowpass filtering algorithm is chosen according to the ratio of the selected cut-off frequency ( $f_c$ ) to the sampling rate ( $f_s$ ), as follows:

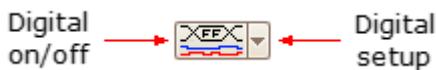
$f_c \div f_s$	Filter type	Description
0.0 to 0.1	Moving average	A moving average filter is used for low cut-off frequencies. The length of the filter is adjusted to achieve the selected cut-off frequency, which is defined as the first minimum in the frequency response. There is significant signal leakage above the cut-off frequency. This filter changes a vertical edge into a linear slope.
0.1 to < 0.5	FIR	A finite impulse response filter is used for medium to high cut-off frequencies. This has a monotonic roll-off above the cut-off frequency and therefore suffers from less leakage than the moving average filter.

You can force PicoScope to use one or other of the filter types by adjusting the **Samples** control in the [Capture Setup toolbar](#) to make the ratio  $f_c/f_s$  fall into one of the two ranges shown in the table. As the table shows, the cut-off frequency must be below half the sampling frequency.

### 7.3.2 Digital Inputs button

Location: [Channels toolbar](#) (MSOs only)

Purpose: controls the settings for the digital inputs of a mixed signal oscilloscope (MSO)



**Digital on/off.** Switches the [digital view](#) on or off. If digital inputs are activated in the [Digital Setup dialog](#), they remain active even when hidden from view.

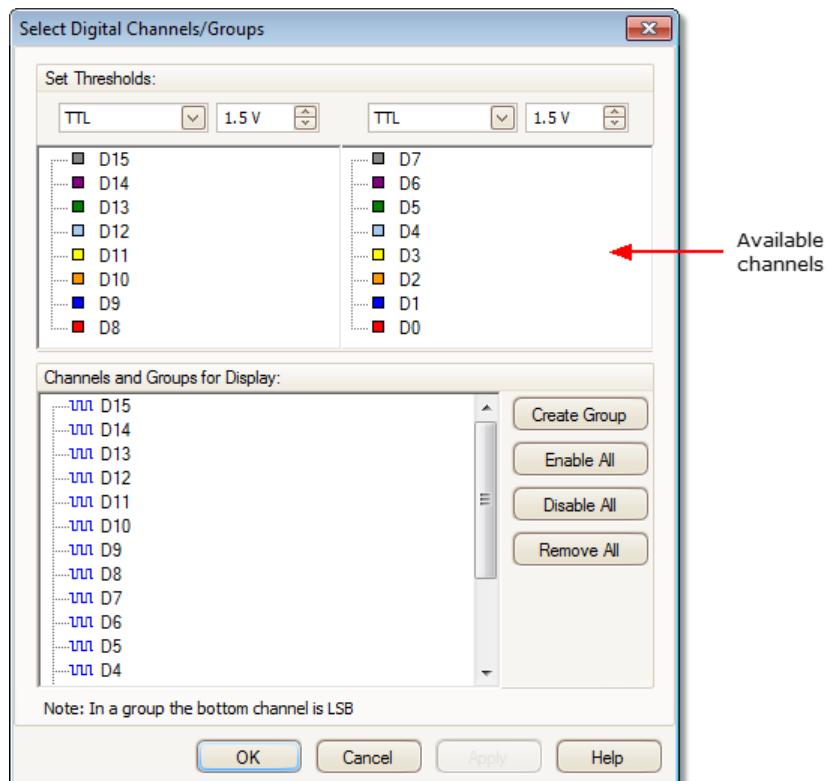


**Digital setup.** Opens the [Digital Setup dialog](#) for channel selection and options.

#### 7.3.2.1 Digital Setup dialog

Location: [MSO button](#)

Purpose: controls the digital inputs of an MSO (mixed signal oscilloscope)



## Set Thresholds

Choose the digital threshold voltage from the drop-down list, or select the **Custom** threshold and set your own voltage using the numeric entry control. The preset thresholds are:

TTL:	1.5 V
CMOS:	2.5 V
ECL:	-1.3 V
PECL:	3.7 V
LVPECL:	2 V
LVCMOS 1.5 V:	750 mV
LVCMOS 1.8 V:	0.9 V
LVCMOS 2.5 V:	1.25 V
LVCMOS 3.3 V:	1.65 V
LVDS:	100 mV
0V Differential:	0 V

Each port has its own independent threshold. Port 0 contains channels D7...D0 and Port 1 contains channels D15...D8.

## Available Channels

This section lists the available digital input channels. They will not be displayed unless you add them to the **Channels and Groups for Display** section of the dialog. Click and drag individual channels into the **Channels and Groups for Display** section, or select a range of channels and drag them all at once, or double-click a channel to add it directly.

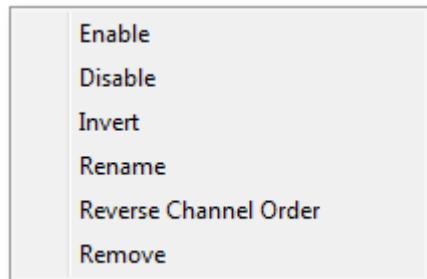
### Channels and Groups for Display

This section lists the digital channels that have been selected for display. Any groups of channels that you have defined will also be listed here.

 indicates a digital channel.

 indicates a group of digital channels. By default, channels added to a group are placed with the most significant bit at the top of the list.

To rename a channel or group, click the name and type. For other operations, right-click the channel or group for a menu of actions:



<b>Enable:</b>	Show the channel. All channels in the list are enabled by default.
<b>Disable:</b>	Hide the channel from display.
<b>Invert:</b>	Reverse the polarity of this channel. Useful for active-low signals.
<b>Rename:</b>	Type a new name for the channel.
<b>Reverse Channel Order:</b>	(Groups only) Reverse the order of channels in the group.
<b>Remove:</b>	Remove channel from the list.

## 7.4 PicoLog 1000 Series Channels toolbar

The **Channels toolbar** controls the settings for each vertical input [channel](#). The toolbar has a different appearance for PicoLog 1000 Series Data Loggers than for PicoScope oscilloscopes (see [Channels toolbar](#) for the standard version).



**Channel control.** This control contains two buttons in one rectangular outline. Click the small triangle on the left to open the [Channel Options dialog](#) with options for [probes](#), [resolution enhancement](#), [scaling](#) and filtering. Click the channel name to toggle the channel on or off.

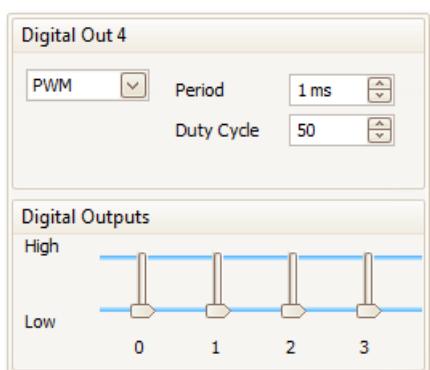


**Digital Outputs button.** For controlling the 2 or 4 digital outputs of the PicoLog 1000 Series device. Opens the [Digital Outputs dialog](#).

### 7.4.1 PicoLog 1000 Series Digital Outputs control

Location: **Digital Outputs button**  on the [Channels toolbar](#)

Purpose: controls the [data logger](#)'s built-in signal generator



**Digital Outputs dialog for the PicoLog 1216**

The range of controls available depends on which model of data logger you have.

#### PWM Output



**PWM.** The PWM output on some devices can be set to generate a pulse-width modulated waveform. This is a logic signal that is toggled with a specified period and duty cycle. The average value of the signal is proportional to its duty cycle, so it can be processed by an external low-pass filter to produce a signal that is proportional to the duty cycle.

**Off:** Disable the PWM output.

**PWM:** Enable the PWM output with the specified controllable **Period** and **Duty Cycle**.



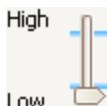
**Period.** Select the duration of one cycle of the PWM output.



**Duty Cycle.** The percentage of the PWM signal period that the signal spends at the logic high level. For example, if the period is 1 ms and the duty cycle 25%, then the signal will spend 25% of 1 ms = 250  $\mu$ s of each cycle at the logic high level, and the remaining 750  $\mu$ s at the logic low level. The voltages of the logic high and low levels are specified in the User's Guide for the data logger, but are typically 0 volts (low) and 3.3 volts (high). Using our example figures, the average value of the PWM output will be 25% x 3.3 volts = 0.825 volts.

#### Digital Outputs

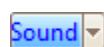
PicoLog PC Data Loggers have one or more digital outputs capable of driving low-current loads.



Each output can be set to a high or low logic level by moving the slider.

## 7.5 USB DrDAQ Channels toolbar

The **Channels toolbar for the USB DrDAQ** controls the settings for each input and output [channel](#):



**Sound waveform sensor control.** The small arrow sets options for the sound waveform input (measured in uncalibrated amplitude units) using the on-board microphone. Click the channel name to toggle the channel on or off.



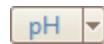
**Sound level sensor control.** The small arrow sets options for the sound level input (measured in decibels) using the on-board microphone. Click the channel name to toggle the channel on or off.



**Scope input control.** The small arrow sets options for the oscilloscope input (the BNC socket marked **Scope**), with options for [probes](#) and [scaling](#). Click the channel name to toggle the channel on or off.



**Resistance input control.** The small arrow sets options for the 0 to 1 MΩ resistance-measuring input on the screw-terminal block. Click the channel name to toggle the channel on or off.



**pH input control.** The small arrow sets options for the pH and ORP (oxidation/reduction potential) measuring input. Click the channel name to toggle the channel on or off.



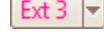
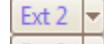
**Temperature sensor control.** The small arrow sets options for the on-board temperature sensor. Click the channel name to toggle the channel on or off.



**Light sensor control.** The small arrow sets options for the on-board light level sensor. Click the channel name to toggle the channel on or off.



**External sensor controls.** The small arrows set options for external sensor inputs 1 to 3. Click the channel name to toggle the channel on or off.



**Signal generator button.** Opens the [Signal Generator dialog](#), which allows you to set the characteristics of the signal generator output.



**RGB LED button.** Opens the [RGB LED Control dialog](#), which allows you to set the color of the on-board LED.

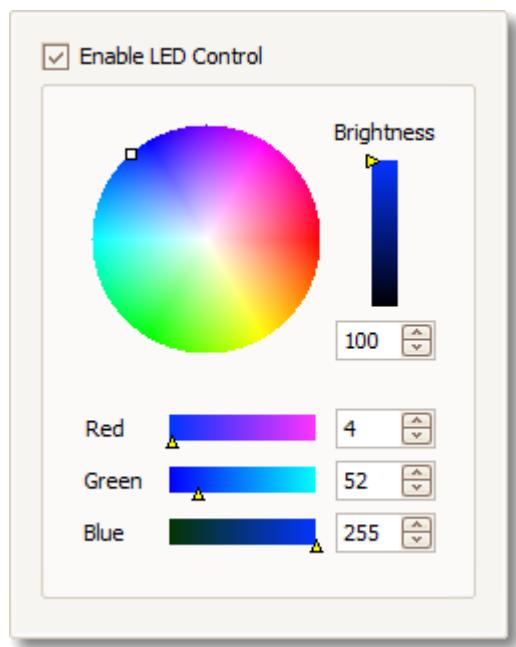


**Digital Outputs button.** Opens the [Digital Outputs dialog](#), which allows you to control the states of the four digital outputs.

### 7.5.1 USB DrDAQ RGB LED control

Location: [USB DrDAQ Channels toolbar](#) > **RGB LED button:** 

Purpose: allows you to set the color of the on-board LED to any one of 16.7 million colors



**Enable LED Control:** **Box checked:** you can set the on-board RGB LED to any color  
**Box clear:** the LED has its normal function of flashing to indicate data capture on the input channels

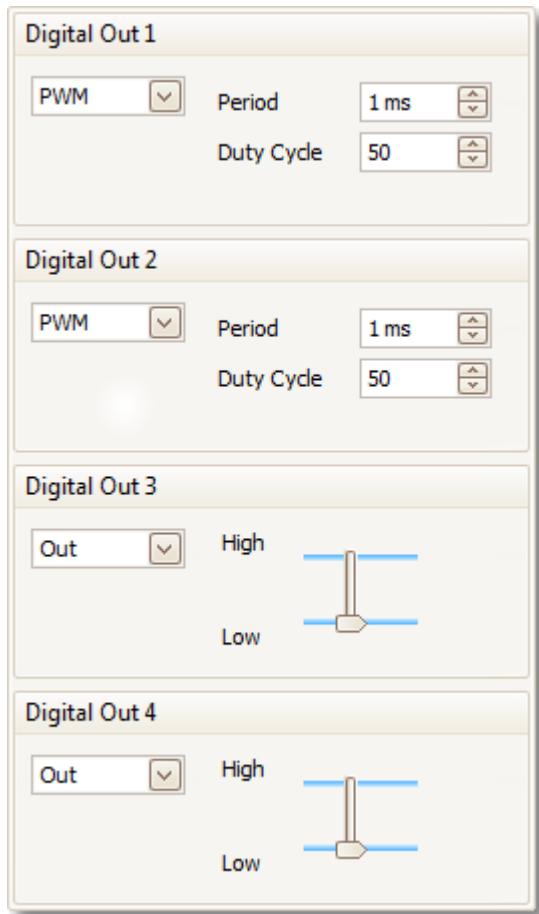
**Other controls:** Experiment with these to see what they do!

### 7.5.2 USB DrDAQ Digital Outputs control

Location: [USB DrDAQ Channels toolbar](#) > **Digital Outputs button:** 

Purpose: allows you to set the characteristics of the four digital outputs on the screw-terminal block.

Each output has its own set of controls:



#### **PWM/Out Control:**

**Set to Out:** you can set the output to either a fixed logic low (near 0 V) or a fixed logic high (near 3.3 V)

**Set to PWM:** the output is a two-level waveform (alternating between 0 V and 3.3 V) with variable **Duty Cycle** and **Period**. The signal can be filtered to produce a DC level proportional to the duty cycle.

#### **Period:**

the time between successive pulses on the output

#### **Duty Cycle:**

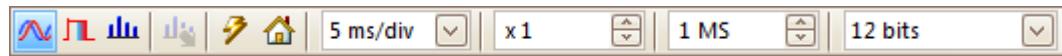
the percentage of the **Period** for which the output is high.

## 7.6 Capture Setup toolbar

The **Capture Setup toolbar** controls the time-related or frequency-related settings of your oscilloscope.

### Scope Mode

In [scope mode](#), the toolbar looks like this:



(See below for different versions of the toolbar in [spectrum mode](#) and [persistence mode](#).)



**Scope Mode.** Sets up PicoScope to operate as an [oscilloscope](#). Use the **Auto Setup button** to optimise the settings. If you wish, you can add a secondary [spectrum view](#) from the context menu (by right-clicking on the scope view).



**Persistence Mode.** Toggles [persistence mode](#), which allows old traces to remain on the screen in faded colors while new traces are drawn on top in brighter colors. The use of colors is controlled by the **Persistence Options dialog**. PicoScope will remember any views that were open, so that you can return to them by clicking the **Persistence Mode** button again.



**Spectrum Mode.** Sets up PicoScope to operate in [spectrum analyzer mode](#). Use the **Auto Setup button** to optimise the settings. If you wish, you can add a secondary [scope view](#) from the context menu (by right-clicking on the scope view).



**Auto Setup.** Searches for a signal on one of the enabled input channels, then sets up the timebase and signal range to display the signal correctly.



**Home.** Restores PicoScope to its default settings. Equivalent to the [File > Startup Settings > Load Startup Settings](#) command.



**Timebase control.** Sets the time represented by a single division of the horizontal axis when the **horizontal zoom control** is set to x1. The timebases available depend on the type of [scope device](#) you are using and, for some devices, on the number and [combination of channels enabled](#) and the selected [trigger mode](#).

Choosing a timebase equal to or longer than PicoScope's *Slow Sampling Transition* setting (200 ms/div by default) causes the scope to switch to a different mode of data transfer. The internal details of this are taken care of by PicoScope, but the slow mode limits the sampling rate to the 'streaming mode sampling rate' specified in the scope device's data sheet. The *Slow Sampling Transition* setting can be changed in the [Tools > Preferences > Sampling](#) dialog.

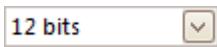
You can change this control to display the total time across the scope view, rather than the time per division, using the **Collection Time Units** control in the [General](#) page of the [Preferences dialog](#).



**Horizontal zoom control.** Zooms the view, in the horizontal direction only, by the specified amount. Click the **+** and **-** buttons to adjust the zoom factor, or the **↶** button to reset.



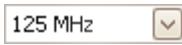
**Samples control.** Sets the maximum number of samples that will be captured for each channel. If this is larger than the number of pixels across the scope view, you can zoom in to see more detail. The actual number of samples captured is displayed on the [Properties sheet](#), and may be different from the number requested here, depending on which timebase is selected and which scope device is in use. To capture one waveform that occupies the entire buffer memory, first set the [Trigger Mode](#) control to **Single**.



**Hardware resolution** ([Flexible Resolution Oscilloscopes](#) only). Sets the number of hardware bits used for sampling. The range of options depends on the number of channels enabled and the sampling rate selected. **Automatic Resolution** chooses the highest resolution compatible with the currently selected sampling rate and capture size. Resolution can be further increased by software filtering: see [resolution enhancement](#).

## Spectrum Mode

In [spectrum mode](#), the **Capture Setup toolbar** looks like this:



**Spectrum Range control.** Sets the frequency range across the horizontal axis of the spectrum analyzer when the **horizontal zoom control** is set to x1.



**Spectrum Options.** Appears if a [spectrum view](#) is open, regardless of whether [scope mode](#) or [spectrum mode](#) is selected. It opens the [Spectrum Options dialog](#).

## Persistence Mode

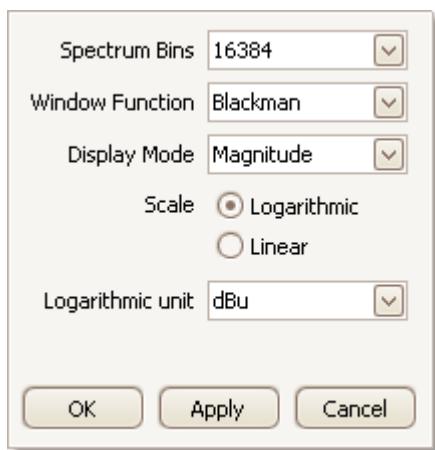
In [persistence mode](#), the **Capture Setup toolbar** looks like this:



**Persistence Options.** Opens the [Persistence Options dialog](#), which controls several parameters affecting the way that PicoScope represents old and new data in persistence mode.

### 7.6.1 Spectrum Options dialog

This dialog appears when you click the **Spectrum Options button** in the [Capture Setup toolbar](#). It is available only when a [spectrum view](#) is open. It contains controls that determine how PicoScope converts the source waveform in the current scope view to a spectrum view.



#### Spectrum Bins

The number of frequency bins into which the spectrum is divided. This control sets the maximum number of frequency bins, which the software may or may not be able to provide depending on other settings. The main constraint is that the number of bins cannot greatly exceed half the number of samples in the source waveform.

If the source waveform contains fewer samples than required (that is, fewer than twice the number of frequency bins), then PicoScope zero-pads the waveform up to the next power of two. For example, if the scope view contains 10,000 samples, and you set Spectrum Bins to 16384, then PicoScope zero-pads the waveform to 16,384 samples, which is the nearest power of two above 10,000. It then uses these 16,384 samples to provide 8,192 frequency bins, not the 16,384 requested.

If the source waveform contains more samples than required, then PicoScope uses as many samples as necessary, starting from the beginning of the waveform buffer. For example, if the source waveform contains 100,000 samples and you request 16,384 frequency bins, PicoScope needs only  $2 \times 16,384 = 32,768$  samples, so it uses the first 32,768 samples from the waveform buffer and ignores the rest. The amount of data actually used is displayed as the **Time Gate** setting in the [Properties sheet](#).

#### Window Function

Allows you to choose one of the standard window functions to reduce the effect of operating on a time-limited waveform. See [Window functions](#).

#### Display Mode

You can choose **Magnitude**, **Average** or **Peak Hold**.

**Magnitude:** the spectrum view shows the frequency spectrum of the last waveform captured, whether live or stored in the [waveform buffer](#).

**Average:** the spectrum view shows a rolling average of spectra calculated from all the waveforms in the [waveform buffer](#). This has the effect of reducing the noise visible in the spectrum view. To clear the averaged data, click [Stop](#) and then [Start](#), or change from **Average** mode to **Magnitude** mode.

**Peak Hold:** the spectrum view shows a rolling maximum of the spectra calculated from all the waveforms in the buffer. In this mode, the amplitude of any frequency band in the spectrum view will either stay the same or increase, but never decrease, over time. To clear the peak hold data, click [Stop](#) and then [Start](#), or change from **Peak Hold** mode to **Magnitude** mode.

Note: when you switch to Average or Peak Hold mode, there may be a noticeable delay while PicoScope processes the entire contents of the waveform buffer, which may contain many waveforms, to build up the initial display. If this occurs, a progress bar appears at the bottom of the window to show that PicoScope is busy:



## Scale

Specifies the labelling and scaling of the vertical (signal) axis. This can be one of the following:

### Linear:

The vertical axis is scaled in volts.

### Logarithmic:

The vertical axis is scaled in decibels, referred to the level selected below in the **Logarithmic unit** control.

**dBV:** Reference level is 1 volt.

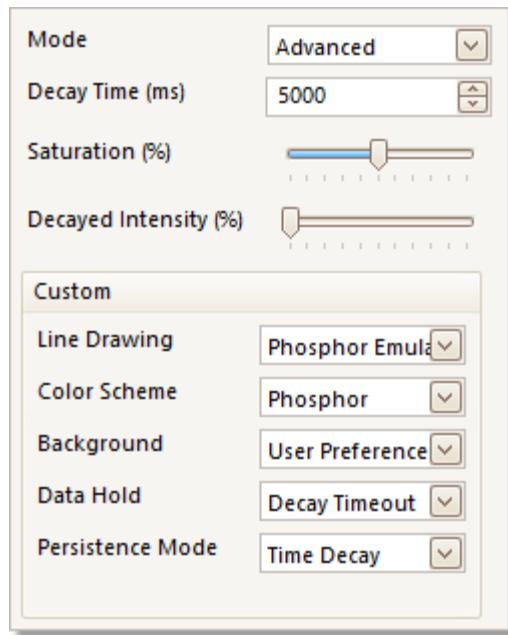
**dBu:** Reference level is 1 milliwatt with a load resistance of 600 ohms. This corresponds to a voltage of about 775 mV.

**dBm:** Reference level is one milliwatt into the specified load impedance. You can enter the load impedance in the box below the **Logarithmic unit** control.

**Arbitrary dB:** Reference level is an arbitrary voltage, which you can specify in the box below the **Logarithmic unit** control.

## 7.6.2 Persistence Options dialog

This dialog appears when you click the **Persistence Options** button  in the **Capture Setup toolbar**. It is available only when [persistence mode](#) is selected. It controls the colors and fading algorithm used to distinguish new or frequent data from old or intermittent data in the persistence view.



### Mode

**Digital Color.** This mode uses a range of colors to indicate the frequency of waveform data. Red is used for the most frequent data, and less frequent data is represented successively by yellow and blue.

**Analog Intensity.** This mode uses color intensity to indicate the age of waveform data. The latest data is drawn at full intensity in the selected color for that channel, with older data being represented by paler shades of the same color.

**Advanced.** This mode opens up a **Custom Options** section at the bottom of the dialog that let you customise the persistence mode display.

### Decay Time

The time, in milliseconds, taken for waveform data to fade from maximum intensity to minimum intensity or from red to blue. The longer the decay time, the longer the older waveforms will remain on the screen.

### Saturation

The intensity or color with which new waveforms are drawn.

### Decayed Intensity

The intensity or color to which the oldest waveforms decay when the decay time expires. If the decayed intensity is zero, then older waveforms will be completely erased from the display after the decay time. For non-zero values of decayed intensity, old waveforms will remain indefinitely on the screen at that intensity unless overwritten by new ones.

### Custom Options

**Line Drawing** The type of line drawn between samples that are adjacent in time.

**Phosphor Emulation.** Joins each pair of sample points with a line whose intensity varies inversely with the slew rate.

**Constant Density.** Joins each pair of sample points with a line of uniform color.

**Scatter.** Draws sample points as unconnected dots.

**Color Scheme** **Phosphor.** Uses a single hue for each channel, with varying intensity.

**Color.** Uses a color from red to blue to represent the age of each waveform.

**Background** **Black.** Overrides the [Color Preferences dialog](#). This is the default.

**White.** Overrides the [Color Preferences dialog](#).

**User Preference.** Sets the background color to the preference set in the [Colors](#) page of the [Preferences dialog](#).

**Data Hold** This option is enabled only when **Persistence Mode** (see below) is set to **Time Delay**.

**Decay Timeout.** Old waveforms fade until they reach **Decayed Intensity** and then disappear.

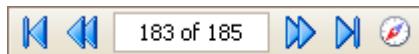
**Infinite.** Old waveforms fade until they reach **Decayed Intensity** and then remain indefinitely unless overwritten by new waveforms.

**Persistence Mode** **Time Delay.** Points on the display are drawn at full intensity when hit by a waveform, and are then allowed to decay to **Decayed Intensity**. The behavior after this depends on the **Data Hold** setting (see above).

**Frequency.** Points on the display are drawn with a color or intensity that depends on the frequency with which they are hit by waveforms.

## 7.7 Buffer Navigation toolbar

The **Buffer Navigation toolbar** allows you to select a waveform from the waveform buffer.



### What is the waveform buffer?

Depending on the settings you have chosen, PicoScope may store more than one waveform in its waveform buffer. When you click the [Start](#) button or change a [capture setting](#), PicoScope clears the buffer and then adds a new waveform to it each time the scope device captures data. This continues until the buffer is full or you click the [Stop](#) button. You can limit the number of waveforms in the buffer to a number between 1 and 10,000 using the [General preferences](#) page.

You can review the waveforms stored in the buffer using these buttons:



**First waveform button.** Display waveform 1.



**Previous waveform button.** Display the previous waveform in the buffer.

1 of 32

**Waveform number indicator.** Show which waveform is currently displayed, and how many waveforms the buffer holds. You can edit the number in the box and press **Enter**, and PicoScope will jump to the specified waveform.



**Next waveform button.** Display the next waveform in the buffer.



**Last waveform button.** Display the last waveform in the buffer.



**Buffer Overview button.** Open the [Buffer Overview window](#) for quick selection of buffer waveforms.

## 7.8 Measurements toolbar

The **Measurements toolbar** controls the [measurements table](#).



It contains the following buttons:

-  **Add Measurement** Adds a row to the table, and then opens the [Add Measurement dialog](#).
-  **Edit Measurement** Opens the [Edit Measurement dialog](#) for the currently selected measurement. You can also edit a measurement by double-clicking on a row of the [measurements table](#).
-  **Delete Measurement** Deletes the currently selected row from the [measurements table](#).
-  **Rulers** Opens the [Ruler settings dialog](#) to control the operation of [phase rulers](#).

This toolbar normally resides at the bottom of the program window, but can be moved to the top using the [Tools > Preferences > Options > Bottom toolbar at top](#) control.

## 7.9 Signal Generator button

The **Signal Generator button** allows you to set up your [scope device's](#) test signal generator, if it has one, or the demo signal settings if PicoScope is in [demo mode](#).



If your scope has a built-in signal generator then clicking the **Signal Generator button** opens the [Signal Generator dialog](#).

If PicoScope is in [demo mode](#) then clicking the **Signal Generator button** opens the [Demo Signals menu](#).

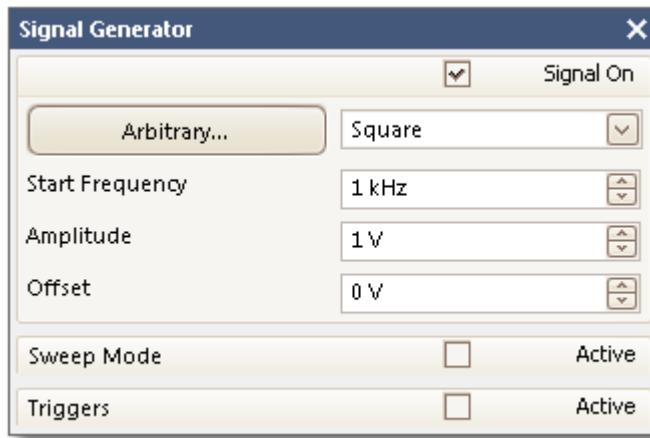
### 7.9.1 Signal Generator dialog (PicoScope devices)

Location: [Signal Generator button](#)  on the toolbar

Purpose: controls the [scope device's](#) built-in signal generator

Not all scope devices have a signal generator, and those that do have a varying range of controls in the signal generator dialog. See the [Device feature table](#) for details.

#### 7.9.1.1 Basic controls



**Signal On.** Tick this box to enable the signal generator.

**Signal Type.** Select the type of signal to be generated. The [list of signal types](#) depends on the capabilities of the scope device.

**Import.** Opens a file selection dialog that allows you to import an [arbitrary waveform file](#). The file will be loaded into the [arbitrary waveform generator](#) and the generator switched on. This button is available only if your scope has an [arbitrary waveform generator](#).

**Arbitrary.** Opens the [Arbitrary Waveform window](#). This button is available only if your scope has an [arbitrary waveform generator](#).

**Start Frequency.** Type in this box or use the spin buttons to select the frequency. If the scope device has a frequency sweep generator, then this box sets the start frequency of the sweep.

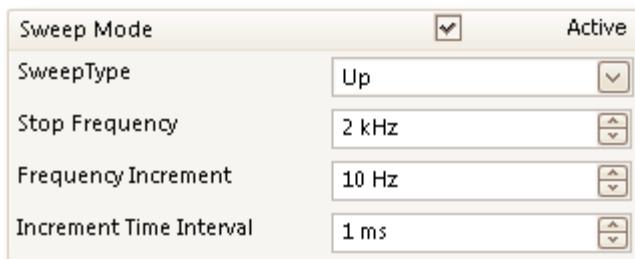
**Amplitude.** The amplitude of the waveform measured from peak to peak. For example, if **Amplitude** is 1 V and **Offset** is 0 V, the output will have a negative peak of -0.5 V and a positive peak of +0.5 V.

**Offset.** The mean value of the signal. For example, when **Offset** is 0 V, a sine or square wave will have equal positive and negative peak voltages.

#### 7.9.1.2 Sweep controls

The signal generator normally generates a fixed frequency set by the **Start**

**Frequency Control.** In sweep mode, it generates a frequency that varies between two specified limits.



**Active.** Tick this box to enable sweep mode.

**Sweep Type.** Specifies the direction in which the frequency is swept.

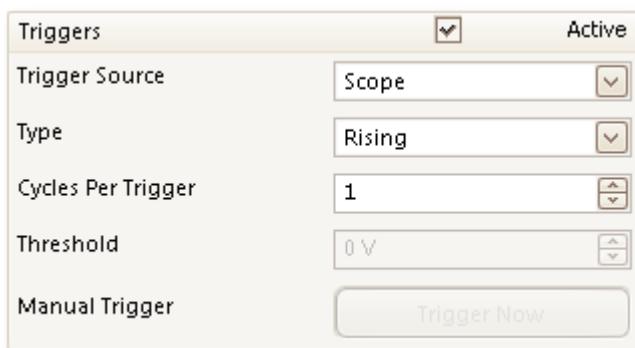
**Stop Frequency.** In sweep mode, the generator stops increasing the frequency when it reaches the **Stop Frequency**.

**Frequency Increment.** In sweep mode, the generator increases or decreases the frequency by this amount every **Increment Time Interval**.

**Increment Time Interval.** In sweep mode, the generator increases or decreases the frequency by **Frequency Increment** each time this interval ends.

#### 7.9.1.3 Trigger controls

The signal generator normally runs continually. If you enable triggering, the signal generator waits for a specified event before generating an output.



**Active.** Tick this box to enable signal generator triggering.

**Trigger Source.** Specifies the signal that will be used to trigger the signal generator:

**Scope.** The same trigger condition that triggers the scope.

**Manual.** The **Trigger Now** button.

**Ext Input.** The input marked **EXT** (if fitted) on the scope device.

An **Alarm** can also be made to trigger the signal generator. This operation is configured in the [Alarms dialog](#).

**Type.** The condition that must be met by the trigger signal:

**Rising.** The signal generator starts running when the trigger signal goes from low to high.

**Falling.** The signal generator starts running when the trigger signal goes from high to low.

**Gate High.** The signal generator runs whenever the trigger signal is high.

**Gate Low.** The signal generator runs whenever the trigger signal is high.

**Cycles per Trigger.** The number of cycles of the specified waveform to generate each time the generator is triggered. If the trigger **Type** is **Gate High** or **Gate Low**, the generator will stop when the gate signal goes inactive even if the requested number of cycles has not been generated.

**Threshold.** Available only when **Trigger Source** is **Ext Input**. Sets the voltage level used to distinguish between the high and low states of the trigger signal.

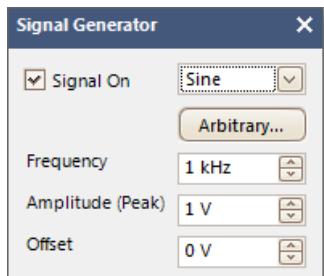
**Manual Trigger.** Available only when **Trigger Source** is **Manual**. Triggers the signal generator to produce the specified number of cycles (or sweeps, if the signal generator is in [sweep mode](#)).

## 7.9.2 Signal Generator dialog (USB DrDAQ)

Location:

[Signal Generator button](#)  on the [USB DrDAQ Channels toolbar](#)

Purpose: controls the USB DrDAQ's built-in signal generator



Signal generator dialog for the USB DrDAQ

### Basic controls



**Signal On.** Tick this box to enable the signal generator.



**Signal Type.** Select the shape of the waveform to be generated.



**Arbitrary.** Opens the [Arbitrary Waveform window](#), allowing you to define your own waveform shape.



**Frequency.** Type in this box or use the spin buttons to select the frequency of the output waveform.



**Amplitude.** The amplitude of the waveform measured from peak to peak. For example, if **Amplitude** is 1 V and **Offset** is 0 V, the output will have a negative peak of -0.5 V and a positive peak of +0.5 V.

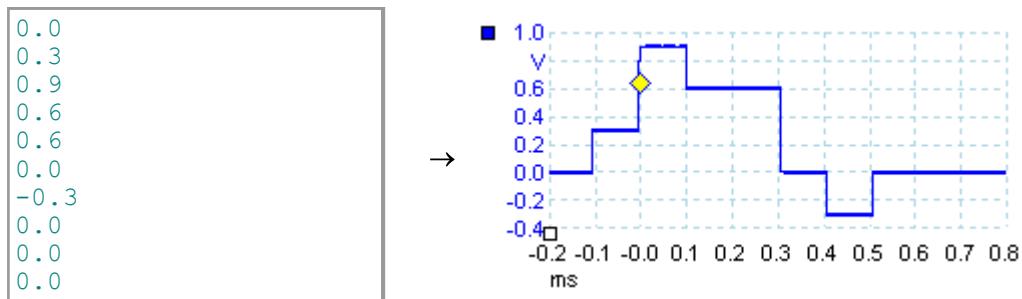


**Offset.** The mean value of the signal. For example, when **Offset** is 0 V, a sine or square wave will have equal positive and negative peak voltages.

### 7.9.3 Arbitrary waveform files

Some PicoScope PC Oscilloscopes have an [arbitrary waveform generator](#) (AWG), which is enabled using the [Signal Generator dialog](#). PicoScope can program the AWG with a standard waveform, such as a sine or square wave, or an arbitrary waveform that you create or import from a text file.

A text file for PicoScope 6 is a list of decimal floating-point values, as in this example:



The file may have between 10 and 8,192 values, as many it needs to define the waveform. Each line may have more than one value, in which case the values must be separated by tabs or commas.

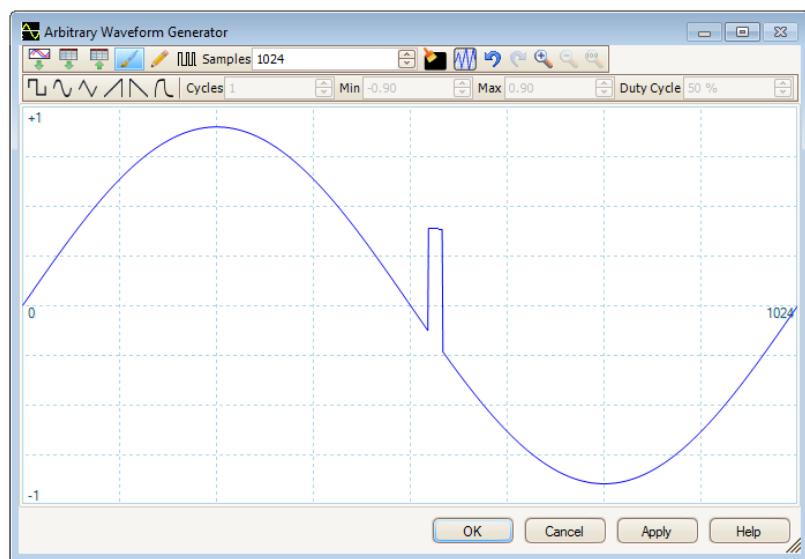
The values are samples between -1.0 and +1.0 and must be equally spaced in time. The output is scaled to the amplitude selected in the [Signal Generator dialog](#), and the selected offset is added if necessary. For example, if the signal generator amplitude is set to "1 V" and the offset to "0 V", then a sample value of -1.0 corresponds to an output of -1.0 V and a sample of +1.0 corresponds to an output of +1.0 V.

The file should contain exactly one cycle of the waveform, which will then be played back at the speed specified in the [Signal Generator dialog](#). In the example above, the signal generator was set to 1 kHz, so one cycle of the waveform lasts for 1 ms. There are 10 samples in the waveform, so each sample lasts for 0.1 ms.

### 7.9.4 Arbitrary Waveform Generator window

Location: [Signal Generator dialog](#) > **Arbitrary**

Purpose: allows you to import, edit, draw and export arbitrary waveforms to load into your scope's [arbitrary waveform generator](#). You can also import and export the data in [CSV format](#) for use in other applications.



Once the desired waveform appears in the window, click **OK** or **Apply** to start using it.

#### Toolbar buttons



**Import from channel.** Opens the [Import from Channel dialog](#), which lets you copy a waveform from the scope into the arbitrary waveform window.



**Import.** Displays an **Open** dialog to allow you to import an arbitrary waveform from a [text file](#).



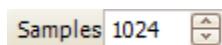
**Export.** Displays a **Save As** dialog to allow you to save the arbitrary waveform as a [text file](#).



**Freehand drawing.** Enters freehand drawing mode. in which you can draw any waveform shape using the mouse.



**Straight-line drawing.** Enters straight line mode, in which you can click on the waveform to draw a straight line from the previous point. To start a new series of lines, click the button again.



**Samples.** The number of samples in the arbitrary waveform. Each sample represents the signal value at a given instant in time, and the samples are equally spaced in time. For example, if there are 1024 samples and the [arbitrary waveform generator](#) is set to play back at 1 kHz, then each sample represents  $(1/1 \text{ kHz} \div 1024)$  or about 0.98 microseconds.



**Bit stream.** Draws a sequence of bits according to binary or hex data that you specify. The logic high and low levels are adjustable.



**Clear.** Deletes the arbitrary waveform.



**Normalize.** Adjusts the waveform vertically so that it occupies the full  $[-1, +1]$  range.



**Undo and Redo.** The **Undo button** reverses the last change made to the arbitrary waveform. The **Redo button** reverses the last action of the **Undo button**.



**Zoom tools.** To zoom the time axis in or out, click the "+" or "-" zoom button and then click on the waveform area. Click the **"100%" button** to restore the time axis to its original scale.

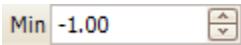
## Waveform settings



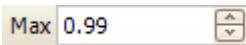
**Standard waveform shapes.** Draw a standard waveform with the settings specified in the numerical controls below the toolbar. The current waveform will be erased.



**Cycles.** The number of cycles to draw. This control is used in conjunction with the **Standard waveform shapes buttons**. Select one of the standard waveform shapes and then enter the number of cycles, and PicoScope will draw the requested number of cycles of the waveform.



**Minimum.** When one of the **Standard waveform shapes buttons** is pressed, this control sets the minimum signal level.



**Maximum.** When one of the **Standard waveform shapes buttons** is pressed, this control sets the maximum signal level.



**Duty cycle.** When a square, triangular or ramp waveform is selected using one of the **Standard waveform shapes buttons**, this control sets the duty cycle of the signal. Duty cycle is defined as the time that the signal spends above zero volts divided by the total cycle time. Thus, a symmetrical square or triangular wave has a duty cycle of 50%. Reducing the duty cycle shortens the positive part of the cycle and lengthens the negative part, and increasing the duty cycle does the opposite.

## Other buttons

### OK

Copies the waveform from the graphical editor into the arbitrary waveform generator and returns to the main [PicoScope](#) window.

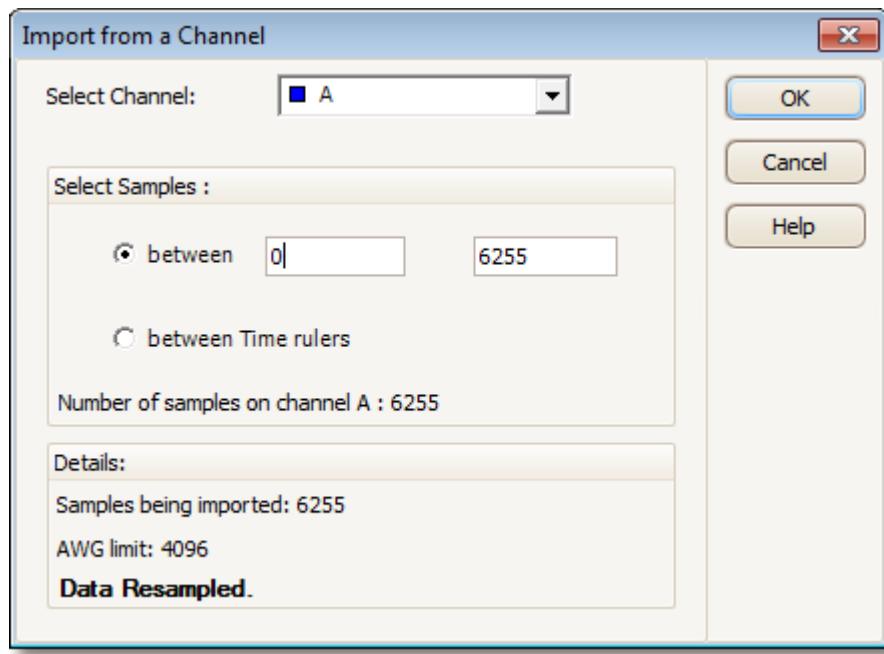
### Apply

Copies the waveform from the graphical editor into the arbitrary waveform generator and remains in the **Arbitrary Waveform Generator window**.

## 7.9.4.1 Import from a Channel dialog

Location: [Arbitrary Waveform window](#) > **Import from a Channel button** (  )

Purpose: allows you to copy captured data from a scope channel to the [Arbitrary Waveform window](#)



**Select Channel:** You can import the latest waveform from any available channel.

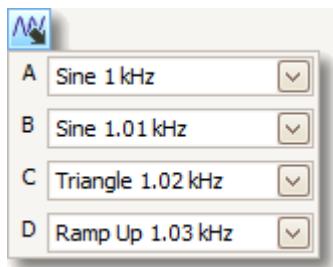
**Select Samples:** By default, the entire capture is imported. This control allows you to specify a subset of the capture, either between specified sample numbers or between rulers. The subset will be scaled to fit the number of samples specified in the **Samples** control in the [Arbitrary Waveform window](#).

### 7.9.5 Demo Signals menu

Location: start PicoScope with no scope connected  
> [Connect Device dialog](#)  
> select ["demo device"](#)  
> [Signal Generator button](#) 

Purpose: allows you to set up test signals so that you can experiment with PicoScope when no scope device is connected

When you click the [Signal Generator button](#) , a drop-down list of all the available channels in the demo device appears, like this:

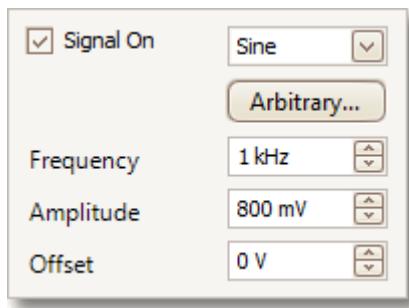


Click one of the channels to open the [Demo Signals dialog](#), which will allow you to set up a signal from that channel.

### 7.9.6 Demo Signals dialog

Location: start PicoScope with no scope device plugged in  
> [Connect Device dialog](#)  
> select "DEMO" device  
> [Signal Generator button](#) (  )  
> select channel

Purpose: controls one channel of the "demo" signal source, a feature of PicoScope that creates a variety of test signals to simulate a scope device



**Signal On:** Tick this box to enable the demo signal source.

**Signal type:** Select from a list of standard signal types.

**Arbitrary Waveform:** Open the [Arbitrary Waveform Editor](#).

**Frequency:** Type your desired frequency in hertz, or use the spin buttons.

**Amplitude:** Type your desired amplitude in volts, or use the spin buttons.

**Offset:** Enter a number to add a d.c. offset to the demo signal. By default, the demo signals have a mean value of zero volts.

## 7.10 Start / Stop toolbar

The **Start / Stop toolbar** allows you to start and stop the [scope device](#). Click anywhere on the toolbar, or press the start/stop key on the keyboard (by default, the space bar), to start or stop sampling.



**Start icon.** Highlighted if the oscilloscope is sampling.



**Stop icon.** Highlighted if the oscilloscope is stopped.

This toolbar normally resides at the bottom of the program window, but can be moved to the top using the [Tools > Preferences > Options > Bottom toolbar at top](#) control.

## 7.11 Triggering toolbar

The **Triggering toolbar** tells the scope device when to start capturing data. See also: [Trigger](#).



**Trigger Mode.** The list of available modes varies depending on the type of [scope device](#) in use.

**None:** PicoScope acquires waveforms repeatedly without waiting for a signal to trigger on.

**Auto:** PicoScope waits for a trigger event before capturing data. If there is no trigger event within a reasonable time, it captures data anyway. It repeats this process until you click the [Stop button](#). "Auto" mode does not set the trigger level automatically.

**Repeat:** PicoScope waits indefinitely for a trigger event before displaying data. It repeats this process until you click the [Stop button](#). If there is no trigger event, PicoScope displays nothing.

**Single:** PicoScope waits once for a trigger event, then stops sampling. To make PicoScope repeat this process, click the [Start](#) button. The **Single** trigger is the only type that allows one capture to fill the entire buffer memory.

**Rapid:** PicoScope instructs the [scope device](#) to acquire a sequence of waveforms with the minimum possible delay between them. The display is not updated until the last waveform in the sequence has been captured. When the operation is finished, you can step through the waveforms using the [Buffer Navigation toolbar](#).

Note: rapid triggering is available only on certain devices (see [device feature table](#)) and on the fastest timebases.

**ETS:** [Equivalent Time Sampling](#). PicoScope captures many cycles of a repetitive signal, then combines the results to produce a single waveform with higher time-resolution than is possible with a single capture. For accurate results, the signal must be perfectly repetitive and the trigger must be stable. ETS is not available on mixed-signal oscilloscopes when digital channels are enabled.

If you select ETS when an [Advanced Trigger](#) type is enabled, the trigger type will revert to **Simple Edge** and the **Advanced Triggering** button will be disabled.



**Advanced Triggering.** Click to open the [Advanced Triggering dialog](#), which gives you extra trigger types beyond the simple edge trigger. If this button is disabled, it is because either **None** or **ETS** is selected in the trigger mode control, or your scope device does not support this mode. To enable the **Advanced Triggering** button, set the control to another trigger mode, such as **Auto**, **Repeat** or **Single**.



**Trigger Source.** This is the channel that PicoScope monitors for the [trigger](#) condition.



**Rising Edge.** Click to trigger on the rising edge of the waveform.

**Falling Edge.** Click to trigger on the falling edge of the waveform.

**Trigger Level.** Sets the [trigger](#) level. You can also set the trigger level by dragging the [trigger marker](#) up or down on the screen.

**Pre-trigger Time** (0% to 100%). This parameter controls how much of the waveform appears before the trigger point. It defaults to 50%, which puts the [trigger marker](#) in the middle of the screen. You can also control this parameter by dragging the [trigger marker](#) to the left or right.

**Post-trigger Delay Enable.** Click this button to toggle the **Post-trigger Delay control** (see next item).

**Post-trigger Delay.** The post-trigger delay is the time that PicoScope waits after the trigger point before sampling. You can also modify this parameter by dragging the [trigger marker](#) while the **Post-trigger Delay button** is enabled. As you drag the marker, you will see the [post-trigger arrow](#) displayed briefly. For this control to have an effect, you must first make sure that the **Post-trigger Delay button** is enabled.

*See the reference topic "[Trigger Timing](#)" for information on how the Pre-trigger Time and Post-trigger Delay controls interact.*

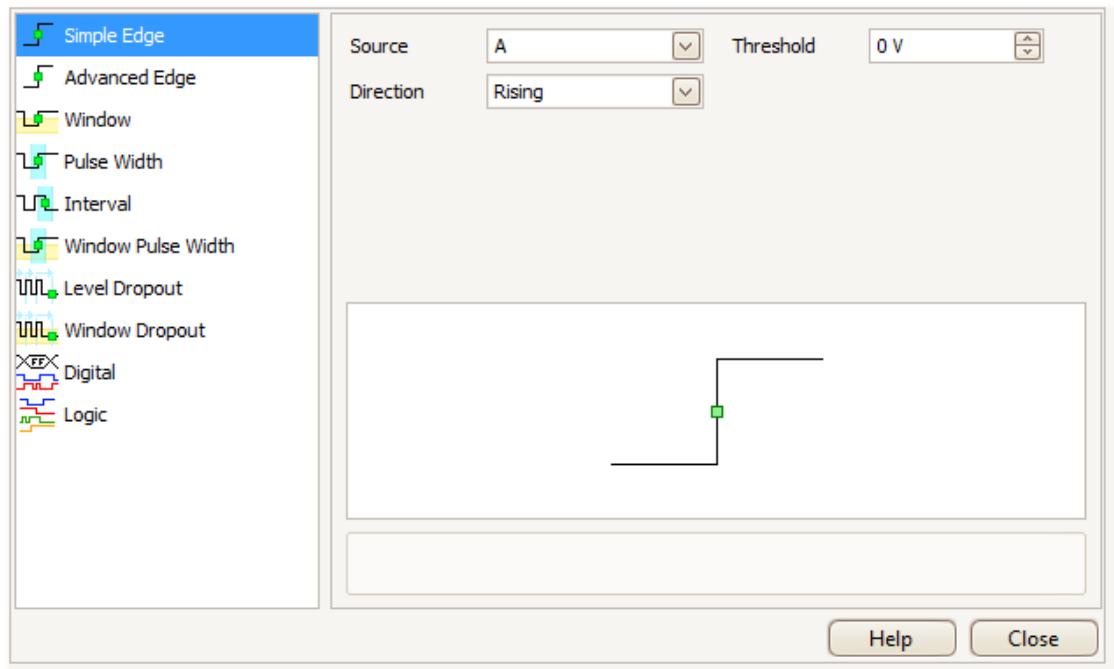
**Rapid Captures.** In [Rapid](#) trigger mode, this is the number of waveforms to capture in a sequence. They will be captured with the minimum possible [dead time](#) between them.

This toolbar normally resides at the bottom of the program window, but can be moved to the top using the **Move Trigger toolbar to top** control in [Tools > Preferences > Options](#).

### 7.11.1 Advanced Triggering dialog

Location: [Triggering toolbar](#) > **Advanced Triggering button** (  )

Purpose: allows you to set up more complex trigger types than simple edge-triggering



**Advanced trigger types list.** This control lists all the available [advanced trigger types](#). Click on the condition you require, and a diagram and description will appear on the right of the dialog.

If [ETS triggering](#) is enabled in the [triggering toolbar](#), then selecting any trigger type except **Simple Edge** switches off ETS mode.



**Advanced triggering options.** The options available depend on the trigger type selected. See [Advanced trigger types](#). Instructions and diagrams also appear in the dialog.

### 7.11.2 Advanced trigger types

The **advanced trigger types** can be switched on in the [Advanced Triggering dialog](#).

For all trigger types except [Digital](#), the first step is to select which signal the scope should use as the trigger; so set **Source** to either **A**, **B**, **Ext** or **AuxIO**. These names correspond to the BNC input connectors on the scope device. Then choose one of the trigger types below.

-  **Simple Edge.** This type provides the same **Rising** and **Falling** edge triggers that are available from the [Triggering toolbar](#). It is included in this dialog as an alternative method of setting up the Simple Edge trigger.

You can set the trigger **Threshold** while in the **advanced triggering dialog**, or alternatively you can drag the [Trigger marker](#) on the scope view.

This is the only trigger type that is compatible with [ETS](#) mode.

-  **Advanced Edge.** This trigger type adds an extra **Rising or Falling** edge trigger, and Hysteresis, to the **Simple Edge** trigger. The **Rising or Falling** option triggers on both edges of a waveform, and is useful for monitoring pulses of both polarities at once. [Hysteresis](#) is described in a separate topic.

-  **Window.** This trigger type detects when the signal enters or leaves a specified voltage window. The **Direction** control specifies whether the trigger should detect the signal entering the window, leaving it, or both. **Threshold 1** and **Threshold 2** are the upper and lower voltage limits of the window. The order in which you specify the two voltages does not matter. [Hysteresis](#) can be set to reduce the number of false triggers on a noisy signal, and is described in a separate topic.

-  **Pulse Width.** This trigger type detects pulses of a specified width.

First set the **Pulse Direction** to either **Positive** or **Negative** according to the polarity of the pulse you are interested in.

Next, set one of the four **Condition** options:

**Greater than** triggers on pulses wider than the specified time.

**Less than** triggers on pulses that are narrower (useful for finding glitches).

**Inside time range** triggers on pulses that are wider than **Time 1** but no wider than **Time 2** (useful for finding pulses that meet a specification).

**Outside time range** does the opposite: it triggers on pulses that are either narrower than **Time 1** or wider than **Time 2** (useful for finding pulses that violate a specification).

Next, set the trigger **Threshold** in volts or other units, or drag the [Trigger marker](#) on the scope view.

Finally, set up **Time 1** (and **Time 2** if present) to define the pulse width.

-  **Interval.** This type lets you search for two successive edges of the same polarity that are separated by a specified interval of time.

First, set the **Starting edge** to either **Rising** or **Falling** according to the polarity of the edges you are interested in.

Next, select one of the four **Condition** options:

**Greater than** triggers when the second edge occurs later than **Time 1** after the first edge (useful for detecting missing events).

**Less than** triggers when the second edge occurs earlier than **Time 1** after the first edge (useful for detecting timing violations and spurious edges).

**Inside time range** triggers when the second edge is later than **Time 1** after the first edge and earlier than **Time 2** (useful for finding valid edges).

**Outside time range** triggers when the second edge is earlier than **Time 1** after the first edge or later than **Time 2** (useful for finding spurious edges).

Finally, set up **Time 1** (and **Time 2** if present) to define the time interval.



**Window pulse width.** This is a combination of the window trigger and the pulse width trigger. It detects when the signal enters or leaves a voltage range for a specified period of time.



**Level dropout.** This detects an edge followed by a specified time with no edges. It is useful for triggering on the end of a pulse train.



**Window dropout.** This is a combination of the window trigger and the dropout trigger. It detects when the signal enters a specified voltage range and stays there for a specified time. This is useful for detecting when a signal gets stuck at a particular voltage.



**Runt.** Detects a pulse that crosses one threshold and then falls below the same threshold, without crossing the second threshold. This is typically used for finding pulses that fail to reach a valid logic level.



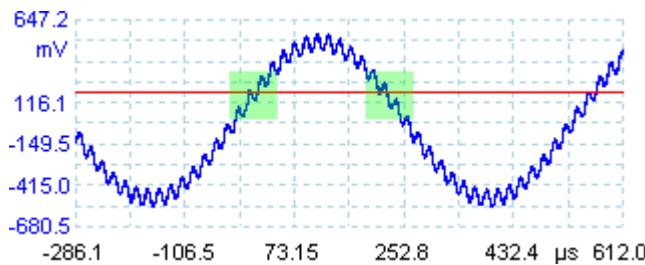
**Digital.** ([MSO devices only](#)) Triggers on a combination of the state of the digital inputs and a transition (edge) on one digital input. See [Digital trigger](#).



**Logic.** This detects a logical combination of the scope's inputs. The conditions that can be applied to each input vary: analog inputs can be edge-, level- or window-qualified; EXT and D15...D0 (if present) are level-qualified with a variable threshold; and AUXIO is level-qualified with a fixed TTL threshold. See [Logic trigger](#).

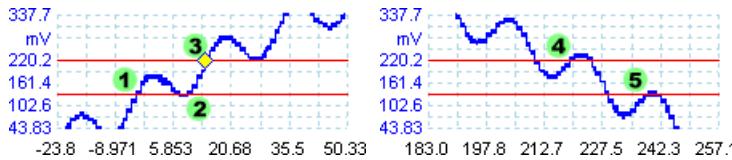
### 7.11.2.1 Hysteresis

**Hysteresis** is a feature of the [advanced trigger types](#) in PicoScope 6 that reduces false triggering on noisy signals. When hysteresis is enabled, a second trigger threshold voltage is used in addition to the main trigger threshold. The trigger fires only when the signal crosses the two thresholds in the correct order. The first threshold arms the trigger, and the second causes it to fire. An example will help to illustrate how this works.



**Noisy signal with a single threshold**

Consider the very noisy signal above. It is difficult to trigger reliably on this signal with a normal rising edge trigger because it crosses the trigger threshold, the red line in this picture, several times in one cycle. If we zoom in on the highlighted parts of the signal, we will see how hysteresis can help.



**Noisy signal with hysteresis threshold**

In these zoomed-in views, the original threshold is the lower red line. The upper red line is the second threshold used by the hysteresis trigger.

The signal rises across the lower threshold at (1) and (2), arming the trigger but not firing it. At (3) the signal finally crosses the upper threshold, firing the trigger. On the falling edge of the signal, at (4) and (5), rising edges of noise pulses cause the signal to cross the upper and lower thresholds, but in the wrong order, so the trigger is not armed and does not fire. Thus the trigger occurs at only one well-defined point in the cycle (3), despite the noise on the signal.

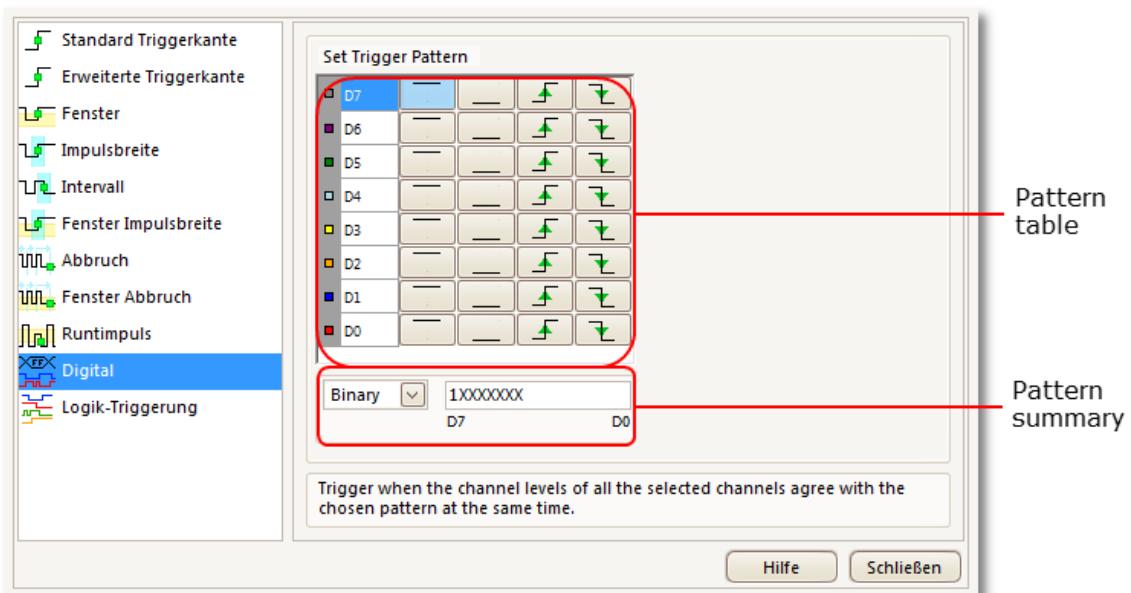
Hysteresis is enabled by default for all the advanced trigger types. The **Hysteresis** controls in the [Advanced triggering dialog](#) let you change the hysteresis voltage as a percentage of full scale. The trigger marker  shows the size of the hysteresis window.

## 7.11.2.2 Digital trigger dialog

Location:

[Advanced triggering dialog](#) > [Digital](#)  and [Logic](#)  buttons

Purpose: sets up triggering on digital inputs

Applicability: [MSO devices](#) only**Pattern table**

Lists all available inputs as selected in the [Digital Setup dialog](#). Each one can be monitored for a low or high level or a rising or falling edge, or ignored. Any number of levels can be specified, but no more than one transition (edge).

<input type="checkbox"/>	D7				
<input type="checkbox"/>	D7				
<input type="checkbox"/>	D7				
<input type="checkbox"/>	D7				
<input type="checkbox"/>	D7				

D7 = X (don't care)  
 D7 = 0 (low level)  
 D7 = 1 (high level)  
 D7 = R (rising edge)  
 D7 = F (falling edge)

**Pattern summary**

This section contains the same settings as the **pattern table** but in a more concise format.

Binary 

The numerical format to use for this section: **Binary** or **Hex(adecimal)**.

The complete triggering pattern and transition. In **Binary** mode the bits are labelled as follows:

RXXXXXXX1010XXXX  
D15 D0

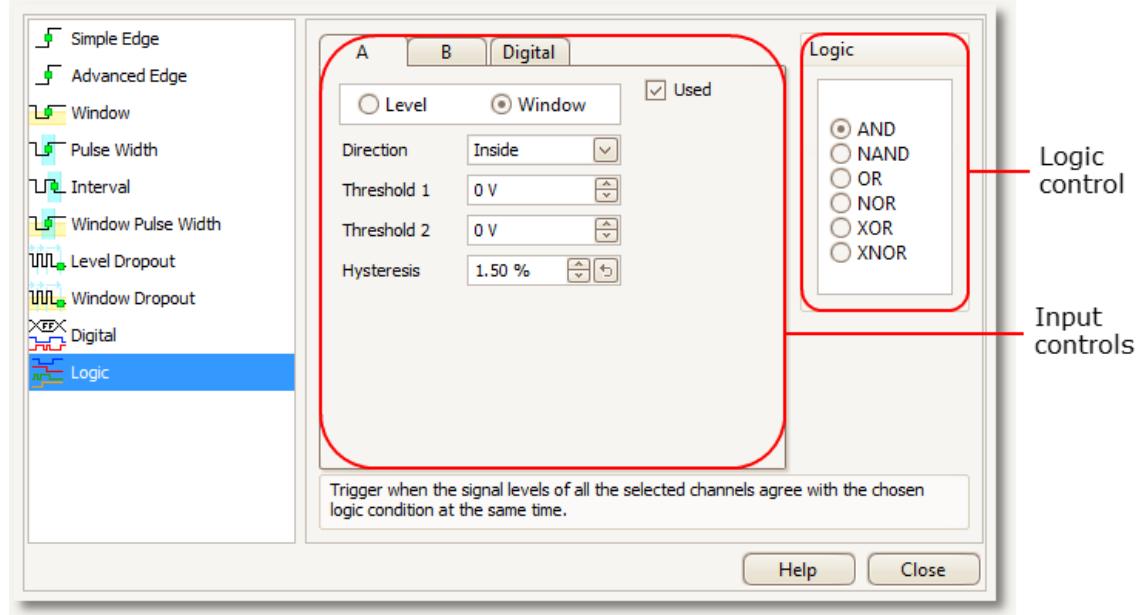
**X** = don't care  
**0** = binary 0  
**1** = binary 1  
**R** = rising edge  
**F** = falling edge

### 7.11.2.3 Logic trigger dialog

Location: [Advanced triggering dialog](#) >  Logic button

Purpose: sets up triggering on a combination of inputs

Applicability: all devices with more than one active input

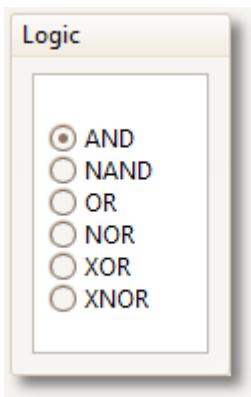


### Input controls

There is a set of controls for each active input of the oscilloscope. The selection of inputs depends on the model of oscilloscope in use. The selection of controls (thresholds, hysteresis, window mode and so on) for each input also depends on the hardware capabilities of the oscilloscope.

<b>A</b>	Channel A
<b>B</b>	Channel B
<b>C</b>	Channel C
<b>D</b>	Channel D
<b>Ext</b>	EXT input ( <a href="#">if present</a> )
<b>AuxIO</b>	AUX input ( <a href="#">if present</a> )
<b>Digital</b>	Digital inputs ( <a href="#">mixed-signal oscilloscopes</a> only). The controls here are the same as those in the <a href="#">Digital Trigger dialog</a> . Check this box to include the relevant input in the Logic Trigger condition. If the box is not checked, the input will be ignored by the Logic Trigger.
<input checked="" type="checkbox"/> <b>Used</b>	

## Logic control



Specifies the Boolean operation used to combine the input trigger conditions. Only inputs with the **'Used'** box checked (see above) are included in the trigger logic.

- AND: all of the input trigger conditions must be met
- NAND: none of the input trigger conditions must be met
- OR: one or more of the input trigger conditions must be met
- NOR: none of the input trigger conditions must be met
- XOR: an odd number of input trigger conditions must be met
- XNOR: an even number of the input trigger conditions must be met

## 7.12 Zooming and Scrolling toolbar

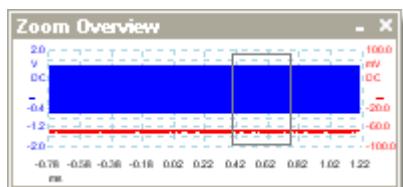
The **Zooming and Scrolling toolbar** allows you to move around a [scope view](#) or [spectrum view](#). Each button has a keyboard shortcut, as listed below.



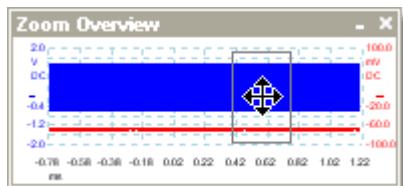
	<b>Ctrl+S</b>	<b>Normal Selection tool.</b> Restores the pointer to its normal appearance. You can use this pointer to click buttons, drag <a href="#">rulers</a> and operate any other controls in the PicoScope window.
	<b>Ctrl+D</b>	<b>Hand tool.</b> Turns the pointer into a hand (☞) that you can use to click and drag the view to pan it vertically and horizontally when you are zoomed in. You can also pan using the scroll bars. Press the <b>Esc</b> key to return to the <b>Normal Selection tool</b> .
	<b>Ctrl+M</b>	<b>Marquee Zoom tool.</b> This button turns the pointer into a marquee zoom tool: (☜). Use it to draw a box (called a marquee) on the view and PicoScope will magnify that box to fill the view. Scroll bars will appear, which you can drag to pan around in the view, or you can pan by using the <b>Hand tool</b> (see above). Zooming in also opens the <a href="#">Zoom Overview</a> window. Press the <b>Esc</b> key to return to the <b>Normal Selection tool</b> .
<p>If you point to the time axis, the pointer changes into the horizontal marquee zoom tool (☜), which restricts zooming to the horizontal axis. This lets you zoom in by an arbitrary amount without disturbing the vertical zoom factor.</p>		
	<b>Ctrl+I</b>	<b>Zoom-in tool.</b> Turns the pointer into a zoom-in tool: (⊕). Click on the view with this tool to zoom in to the specified location. Zooming in also opens the <a href="#">Zoom Overview</a> window.
<p>If you point to the time axis, the pointer changes into the horizontal zoom-in tool (⊕), which restricts zooming to the horizontal axis. This lets you zoom in without disturbing the vertical zoom factor.</p>		
	<b>Ctrl+O</b>	<b>Zoom-out tool.</b> Turns the pointer into a zoom-out tool: (⊖). Click on the view with this tool to zoom out around the specified location.
<p>If you point to the time axis, the pointer changes into the horizontal zoom-out tool (⊖), which restricts zooming to the horizontal axis. This lets you zoom out without disturbing the vertical zoom factor.</p>		
	<b>Undo zoom.</b> Returns the view to the previous zoom and pan settings.	
	<b>Ctrl+U</b>	<b>Zoom to full view.</b> Resets the view to normal size. The view will no longer have scroll bars, and panning will no longer be possible.

### 7.12.1 Zoom Overview

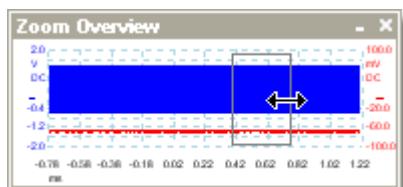
Whenever you zoom in using the [Zooming and Scrolling toolbar](#), the **Zoom Overview** window should appear\*:



The **Zoom Overview** shows the full waveforms on all enabled channels. The rectangle indicates the area that is visible in the current view.



You can move around the waveform by dragging the rectangle.



You can also adjust the zoom factor by dragging the edges of the rectangle to resize it.



**Minimize** button: reduce the **Zoom Overview** window in size without affecting the zoom settings.



**Close** button: close the **Zoom Overview** window and return the zoom factor to 100%.

\*Note: if the **Zoom Overview** does not appear, the feature may have been switched off. Check the **Zoom Overview** option in [Tools > Preferences > Options](#).

## 8 How to...

This chapter explains how to perform some common tasks.

### 8.1 How to change to a different device

- ➊ Unplug the old [device](#).
- ➋ Cancel the **Check USB cable** dialog.
- ➌ Plug in the new device.
- ➍ PicoScope will detect the new device and begin to use it. If more than one device is connected, PicoScope will ask you which one to use.

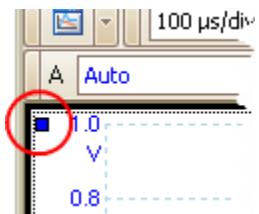
## 8.2 How to use rulers to measure a signal

### Using a single ruler for signal-to-ground measurements

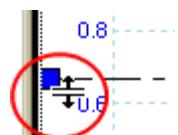
- Look at the [Channels toolbar](#) to find the color code for the [channel](#) you wish to measure:



- Find the ruler handle (the small colored square in the top-left or top-right corner of the [scope view](#) or [spectrum view](#)) of this color:



- Drag the ruler handle downwards. A [signal ruler](#) (horizontal broken line) will appear across the view. Release the ruler handle when the ruler is where you want it.



- Look at the [ruler legend](#) (the small table that appears on the view). It should have a row marked by a small colored square matching the color of your ruler handle. The first column shows the signal level of the ruler.

1	2	▲	▼
■ 586.0mV	---	---	---

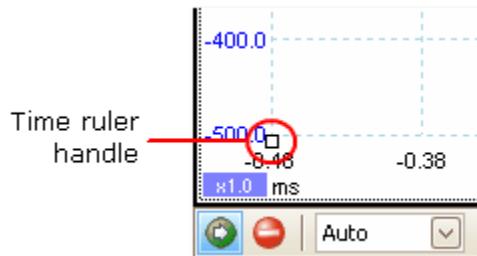
### Using two rulers for differential measurements

- Follow the steps above for "using a single ruler".
- Drag the second ruler handle of the same color downwards until its ruler is at the signal level to be measured.
- Look at the [ruler legend](#) again. The second column now shows the signal level of the second ruler, and the third column shows the difference between the two rulers.

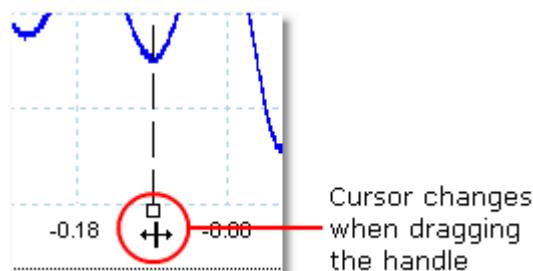
1	2	▲	▼
■ 586.0mV	-493.0mV	1.079V	-

### 8.3 How to measure a time difference

- Find the time ruler handle (the small white square in the bottom left corner of the [scope view](#)).



- Drag the ruler handle to the right. A [time ruler](#) (vertical broken line) will appear on the scope view. Release the ruler handle when the ruler is at the time you wish to use as the reference.



- Drag the second white ruler handle to the right until its ruler is at the time to be measured.
- Look at the [ruler legend](#) (the small table that appears on the scope view). It should have a row marked by a small white square. The first two columns show the times of the two rulers, and the third column shows the time difference.

1	2	Δ	-
□ -129.0 $\mu$ s	44.0 $\mu$ s	85.0 $\mu$ s	

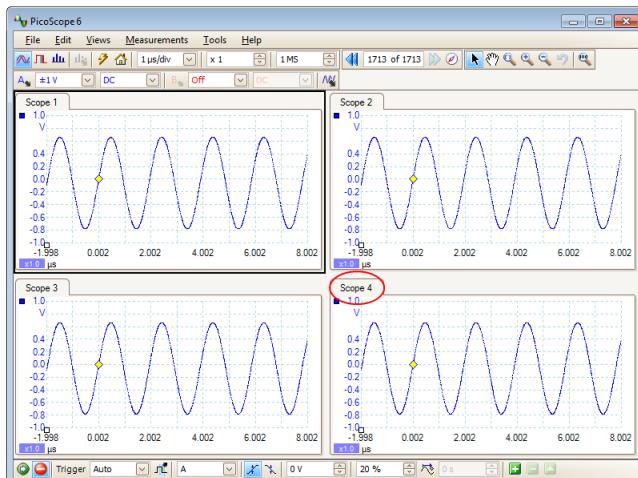
- The [frequency legend](#) shows  $1/\Delta$ , where  $\Delta$  is the time difference.

□ 1/Δ 33.37 Hz, 2002.0 RPM

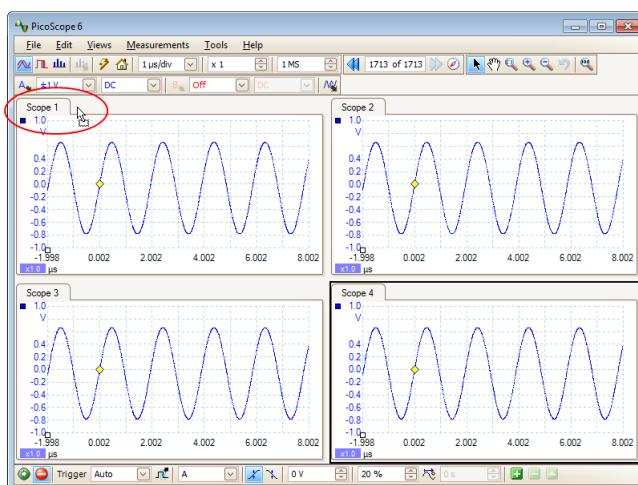
You can use a similar method to measure a frequency difference on a [spectrum view](#).

## 8.4 How to move a view

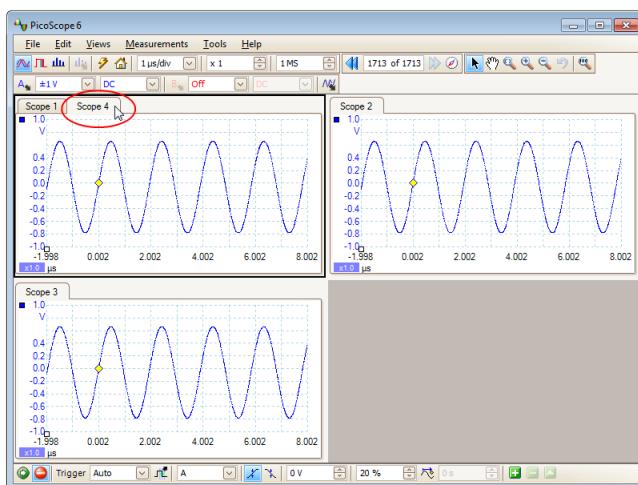
You can easily drag a [view](#) from one [viewport](#) to another. This example shows four viewports, which contain [scope views](#) called "Scope 1" to "Scope 4". Suppose that you wish to move the "Scope 4" view to the top left viewport.



1. Click on the name tab of the "Scope 4" view and hold the mouse button down.



2. Drag the mouse pointer to the new location next to the name tab of the "Scope 1" view.



3. Release the mouse button, and the view will move to the new location.



## 8.5 How to scale and offset a signal

PicoScope offers several ways to change the size and position of a signal during or after capture. These methods apply equally to [scope views](#) and [spectrum views](#). They do not change the stored data, only the way in which it is displayed. These options are provided in addition to the [analog offset](#) capability of some scopes (see [Device feature table](#)).

### Global zooming and scrolling

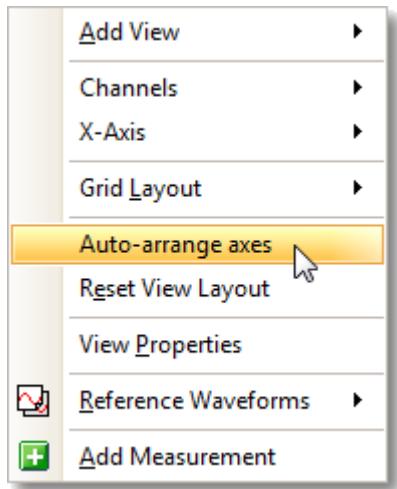
This is usually the quickest way to get a closer look at the fine detail on your signals. The global zooming and scrolling tools move all the signals at once and are found on the [zooming and scrolling toolbar](#).



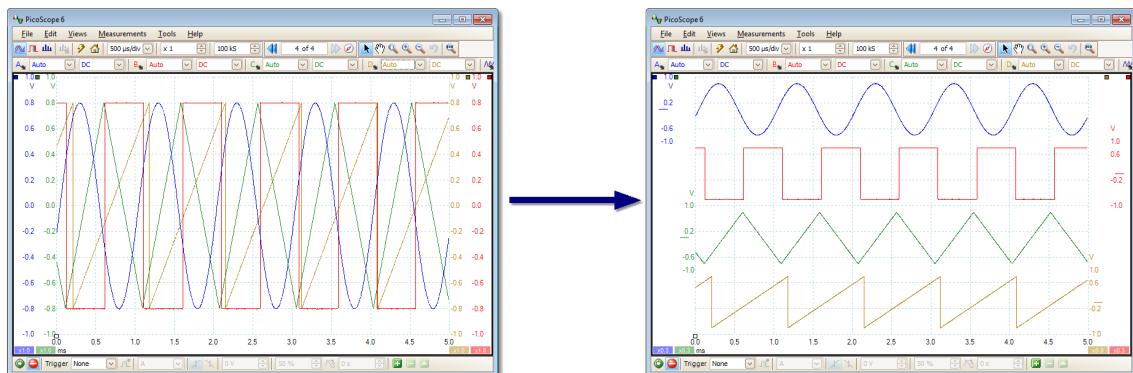
When a view is zoomed in, it has vertical and horizontal scroll bars that let you move the signals around as a group. You can also use the hand tool to scroll around the graph.

### Auto-arrange axes

Right-click the scope or spectrum view and select [Auto-arrange axes](#):



PicoScope automatically scales and offsets the channels to make them fit on the view without overlapping. This is the quickest way to tidy up the scope view:



## Axis scaling and offset

Use these tools if **Auto-arrange axes** (see above) does not give you the results you want. It allows you to position channels individually on the view (unlike the global zooming and scrolling tools, which are applied to all of the channels at the same time).

Click the scaling button  at the bottom of the axis you wish to modify, and the [axis scaling controls](#) will appear. To adjust the offset without using the axis scaling controls, click on the vertical axis and drag it up or down.

## How is this different from scaling my data with a Custom Probe?

New Probe...

You can create a [Custom Probe](#) to apply scaling to the raw data. A Custom Probe may change the scale and position of data on the graph but it has a few important differences from the other scaling methods.

- Custom Probe scaling is a permanent transformation. The scaling is applied when the waveform is captured and cannot be changed afterwards.
- The actual data values themselves are changed, so the graph axes may no longer display the original voltage range of the device.
- Custom Probe scaling can be nonlinear and so may alter the shape of the signal.

Custom Probes are useful when you want to represent the characteristics of a physical probe or transducer that you plug into your scope device. All of the zooming, scrolling, scaling and offset tools still apply to data that has been scaled with a Custom Probe in exactly the same way that they would apply to the raw data.

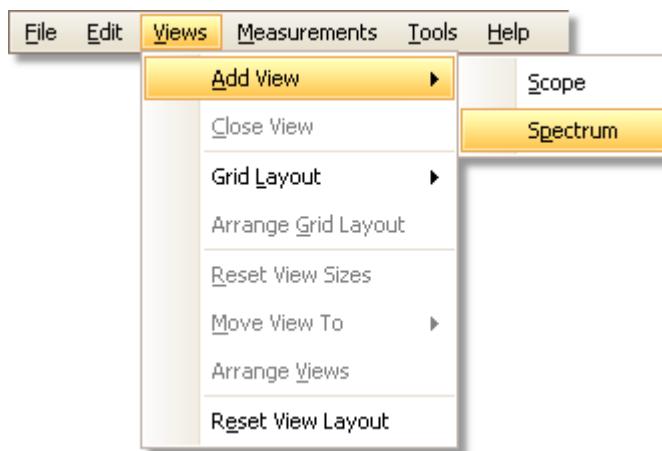
## 8.6 How to set up the spectrum view

### Creating a spectrum view

First, ensure that the [trigger mode](#) is not set to [ETS](#), as it is not possible to open a spectrum view in ETS trigger mode.

There are three ways to open a [spectrum view](#):

- Click the **Spectrum Mode button** in the [Capture Setup toolbar](#). We recommend using this method to get the best spectrum analysis performance from your scope. Once in Spectrum Mode, you can still open a scope view to see your data in the time domain, but PicoScope optimizes the settings for the spectrum view.
- Go to the [Views menu](#) select **Add view**, then select **Spectrum**.



This method opens a spectrum view in the currently selected mode, whether this is Scope Mode or Spectrum Mode. For best results, we recommend that you switch to Spectrum Mode, as described in the method immediately above.

- Right-click on any [view](#), select **Add view**, then select **Spectrum**. The menu is similar to the [Views menu](#) shown above.

### Configuring the spectrum view

See [Spectrum Settings dialog](#).

### Selecting the source data

PicoScope can produce a [spectrum view](#) based on either live or stored data. If PicoScope is running (the [Start](#) button is pressed in), the spectrum view represents live data. Otherwise, with PicoScope stopped (the [Stop](#) button pressed in), the view represents data stored in the currently selected page of the waveform buffer. When PicoScope is stopped, you can use the [buffer controls](#) to scroll through the buffer and the spectrum view will be recalculated from the waveform currently selected.

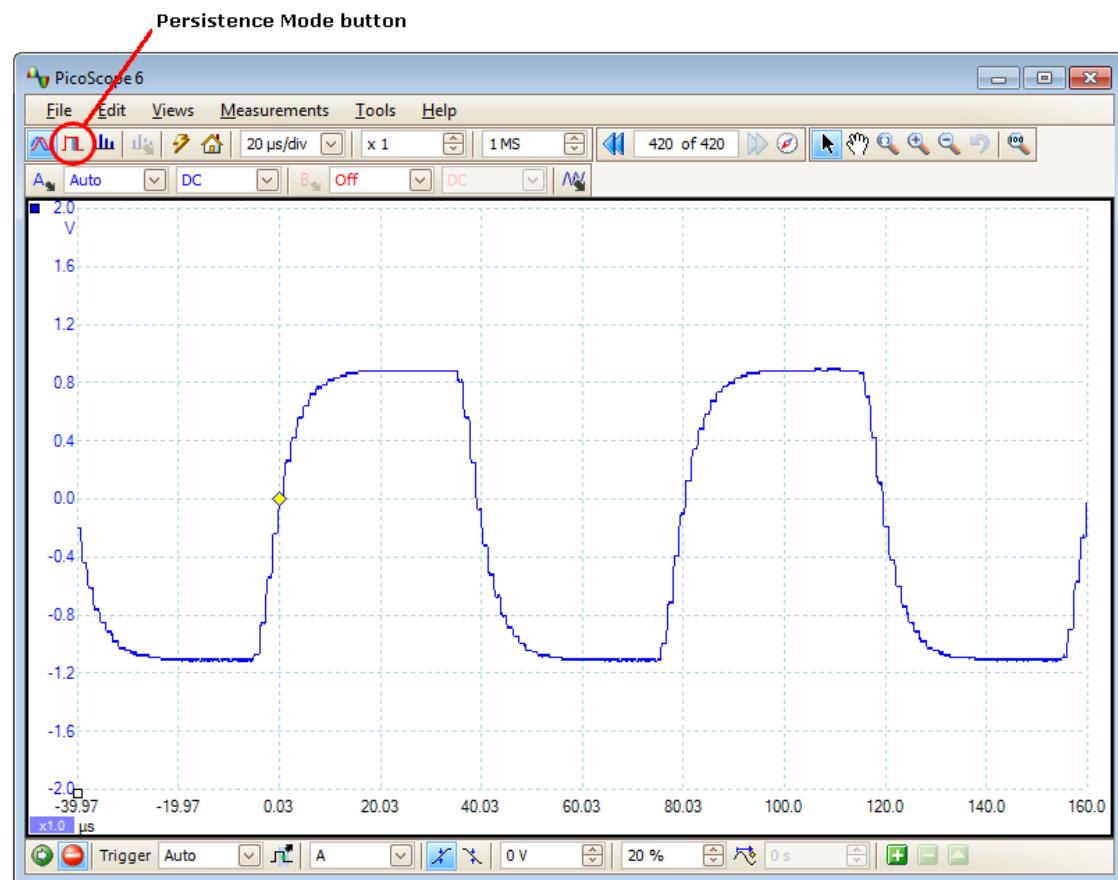
## 8.7 How to find a glitch using persistence mode

**Persistence mode** helps you find rare events hidden in otherwise repetitive waveforms. In normal scope mode, such an event may appear on the display for a fraction of a second, too quickly for you to press the space bar to freeze it on the screen. Persistence mode keeps the event on the display for a predetermined time, allowing you to set up the trigger options to capture it more reliably.

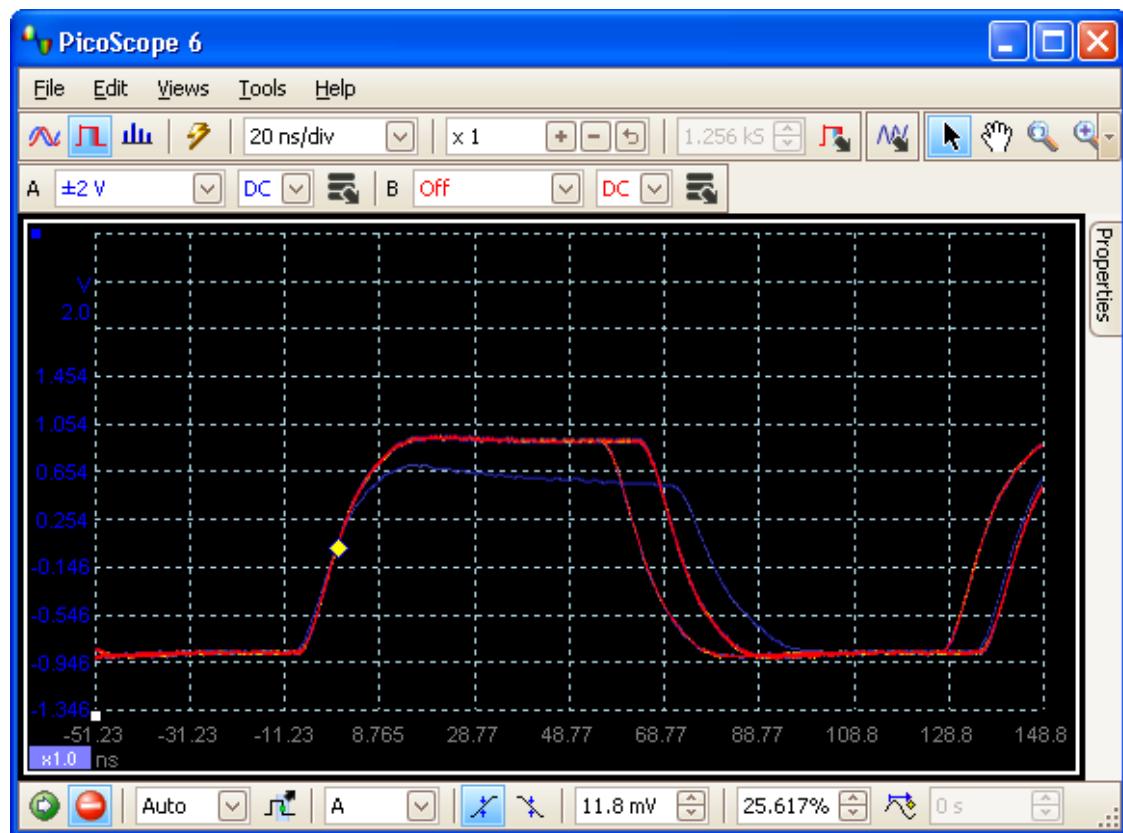
### Step-by-step guide

- Set up the scope to trigger on a repetitive waveform like the one below. We suspect that there are occasional glitches but we can see nothing wrong yet, so we shall use persistence mode to investigate.

Click the **Persistence Mode button** to continue.

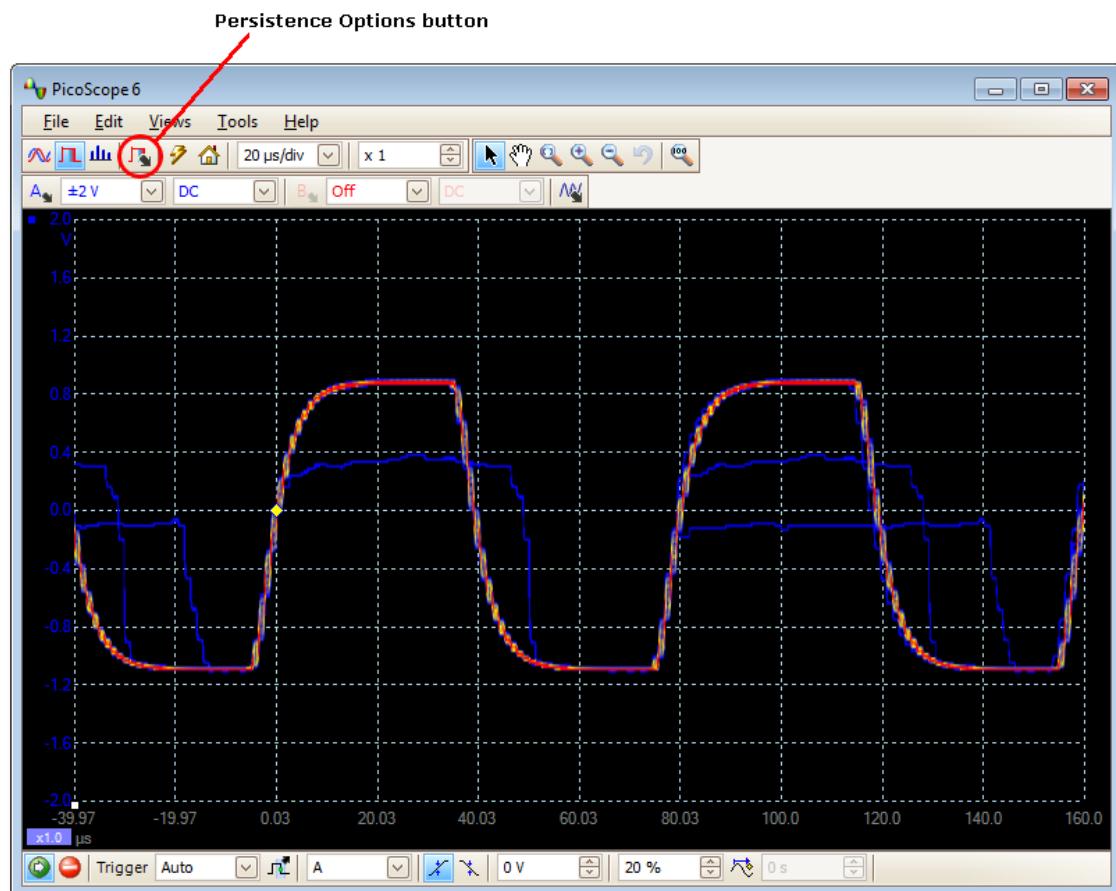


- Our original scope view is replaced by a persistence view, as shown below. Immediately, we can see three pulses with different shapes. At this point we have the **Saturation** control in [Persistence Options](#) turned up to maximum to help us spot the various waveforms easily.



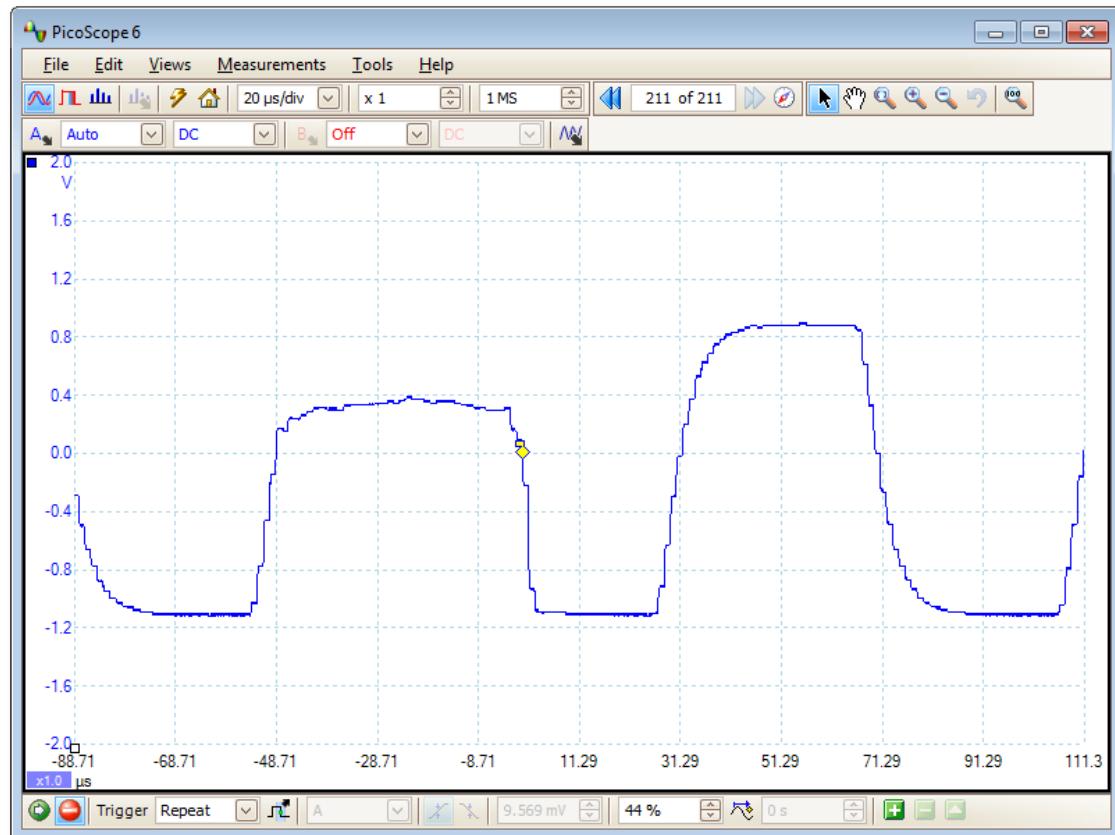
- Now that we have found some glitches, we will turn the **Saturation** control down to minimum. Click the **Persistence options button** to open the [Persistence Options dialog](#), and then use the slider to adjust the saturation. The display then appears as below.

The waveforms are now darker but have a wider range of colors and shades. The most frequently occurring waveform is shown in red, and is the normal shape of the pulse. A second waveform is drawn in light blue to shows that it occurs less frequently, and it shows us that there is an occasional jitter of about 10 ns in the pulse width. The third waveform is drawn in dark blue because it occurs less frequently than the other two, and indicates that there is an occasional runt pulse about 300 mV lower in amplitude than normal.



- Persistence mode has done its job. We have found our glitches, and now we want to examine them in more detail. The best way to do this is to switch back to normal [scope mode](#), so that we can use the [advanced triggering](#) and [automatic measurement](#) functions built in to PicoScope.

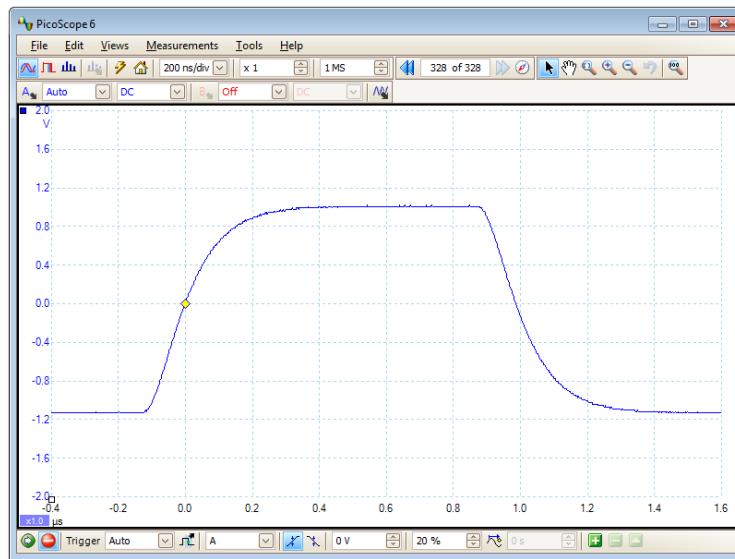
Click the Scope Mode button. Set up an advanced pulse-width trigger to look for a pulse wider than 60 ns. PicoScope then finds the runt pulse straight away.



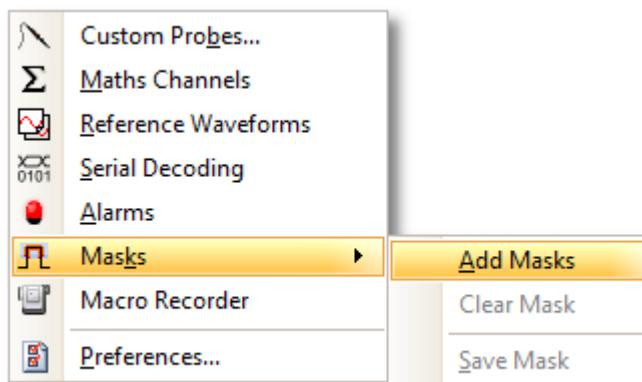
We can now add automatic measurements or drag the rulers into place to analyze the runt pulse in detail.

## 8.8 How to set up a Mask Limit Test

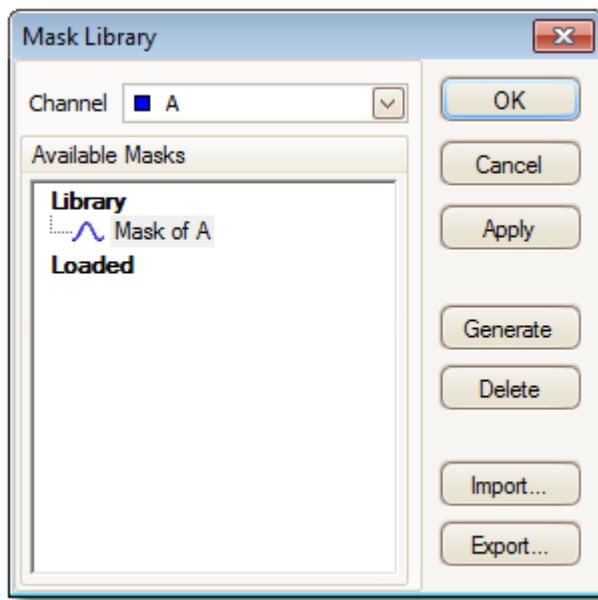
1. Display a stable waveform in a [scope view](#). Adjust the voltage range and timebase so that the feature of interest fills most of the view. In this example, we are viewing a repetitive pulse as might be found on a data bus.



2. Select the [Tools](#) > [Masks](#) > [Add Masks](#) command.

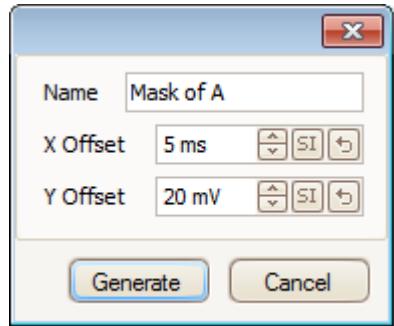


3. You should now be in the [\*\*Mask Library dialog\*\*](#):

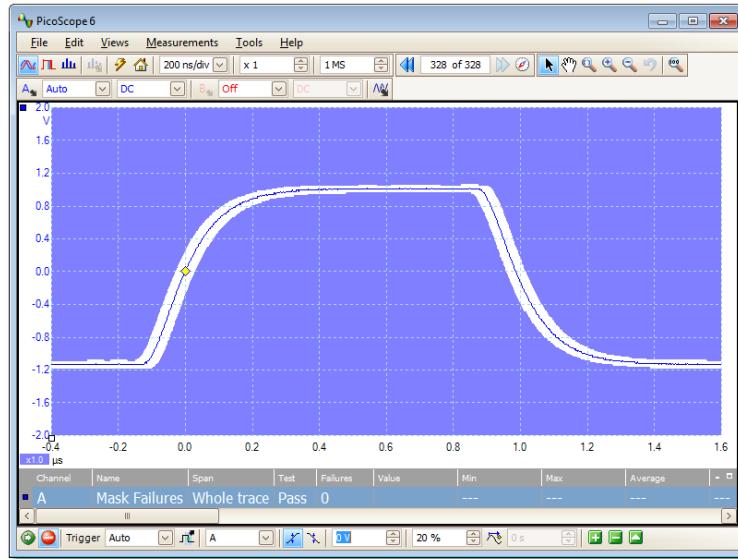


Channel A is selected by default. You can change this if you want to apply the mask to a different channel.

4. Click the **Generate** button to open the [\*\*Generate Mask dialog\*\*](#):

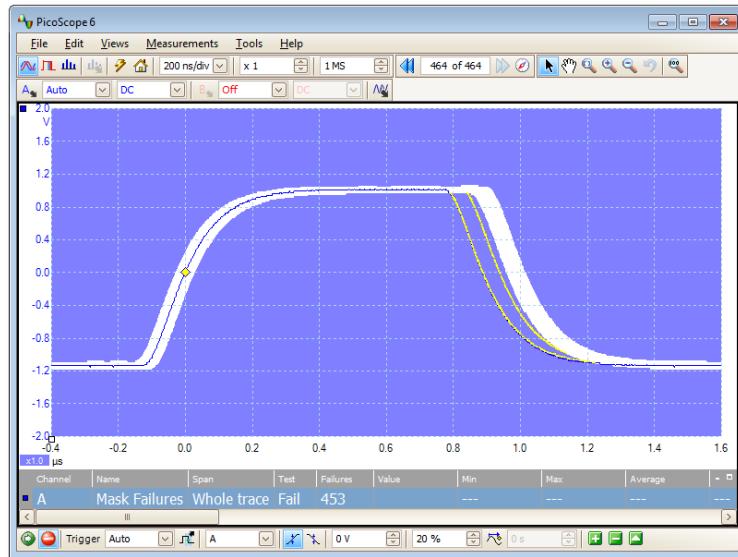


5. For now, accept the default settings and click **Generate**. Then click **OK** in the [Mask Library dialog](#) to return to the scope view:



You now have a mask drawn around the original waveform.

6. PicoScope stops capturing when you enter the [Mask Library dialog](#), so press the space bar to restart. If any captured waveform fails to fit inside the mask, the offending parts are drawn in a contrasting color. The [Measurements table](#) shows the number of failures:



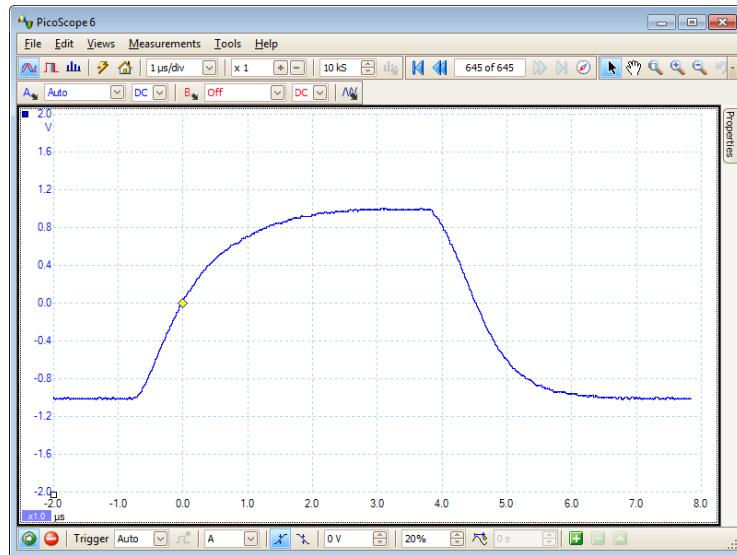
7. You now have a functioning mask limit test. Please read the [Mask Limit Test](#) topic for information on editing, importing and exporting masks. It is also possible to set up a Mask Limit Test on a [spectrum](#) or [XY](#) view.

For further information on this feature, see: [Mask Limit Testing](#).

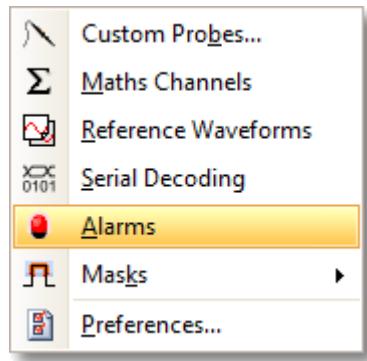
## 8.9 How to save on trigger

**Save-on-trigger** is just one of a number of functions that are possible with the [Alarms](#) feature.

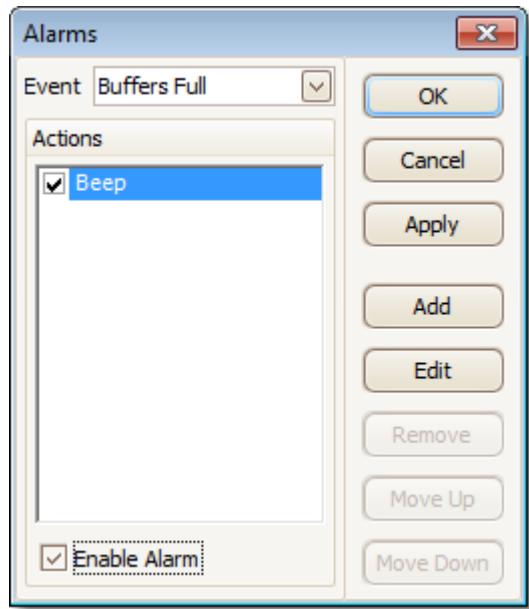
1. Set up PicoScope to display your waveform, and enable triggering:



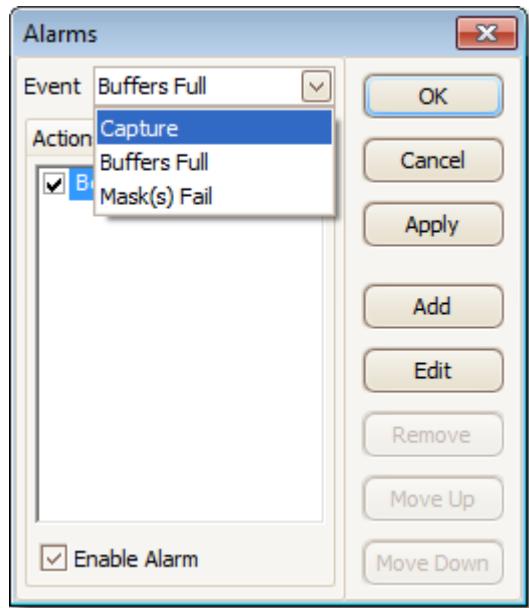
2. Select the [Tools](#) > [Alarms](#) command:



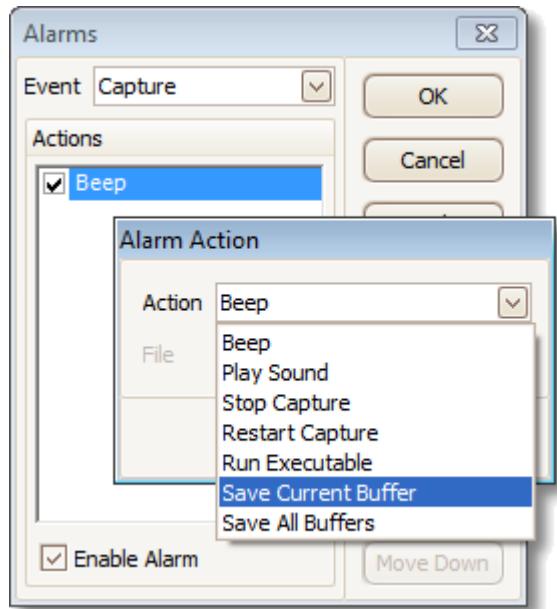
3. You should now be in the [Alarms dialog](#):



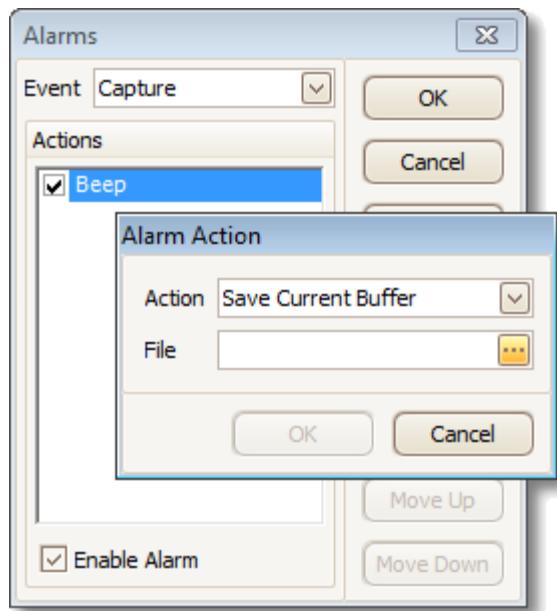
4. Set **Event** to **Capture**:



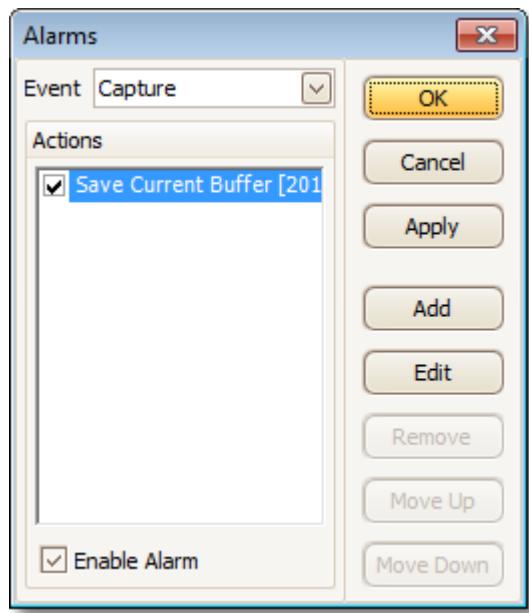
5. Select the first item in the **Actions** list, click **Edit**, and change **Action** to **Save Current Buffer**:



6. Click the  button to the right of the **File** box and enter the name and location of the file to save:



7. Ensure that both the **Save Current Buffer** check box and the **Enable Alarm** check box are set:

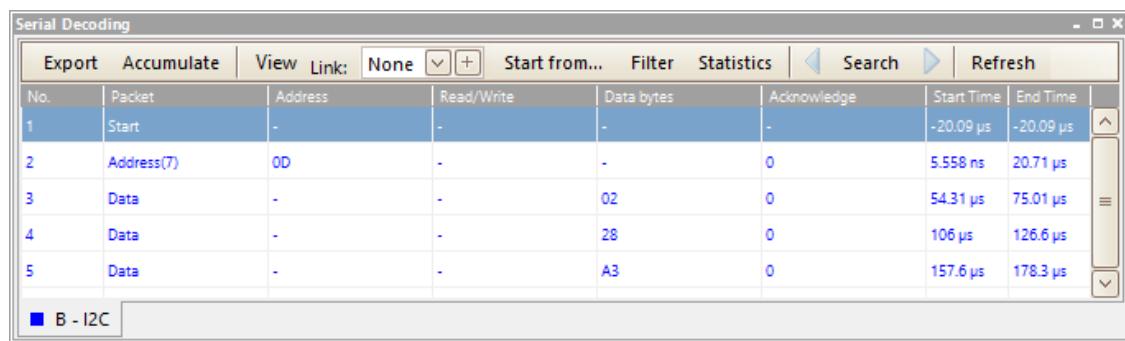


7. Click **OK**. PicoScope will now save a file on every trigger event.
8. Switch off the alarm when you have finished using it, to avoid creating unwanted files.

## 8.10 How to create a link file

A **link file** is used by the [Serial Decoding "In Table" view](#) to convert numerical data to text. This is useful for displaying decoded packets in a human-readable format.

1. In the **Tools > Serial Decoding** menu, enable serial decoding and select the required serial protocol. In this example we shall use I<sup>2</sup>C. Ensure that the **In Table** view is enabled.
2. PicoScope will show the decoded data in numeric format:



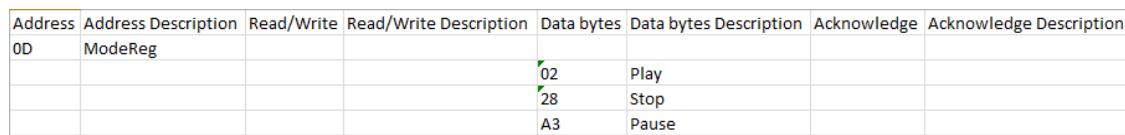
The screenshot shows the 'Serial Decoding' window with the 'B - I2C' protocol selected. The data table has columns for No., Packet, Address, Read/Write, Data bytes, Acknowledge, Start Time, and End Time. The data rows are: 1. Start, Address: -, Read/Write: -, Data bytes: -, Acknowledge: -, Start Time: -20.09 µs, End Time: -20.09 µs; 2. Address(7), Address: 0D, Read/Write: -, Data bytes: -, Acknowledge: 0, Start Time: 5.558 ns, End Time: 20.71 µs; 3. Data, Address: -, Read/Write: -, Data bytes: 02, Acknowledge: 0, Start Time: 54.31 µs, End Time: 75.01 µs; 4. Data, Address: -, Read/Write: -, Data bytes: 28, Acknowledge: 0, Start Time: 106 µs, End Time: 126.6 µs; 5. Data, Address: -, Read/Write: -, Data bytes: A3, Acknowledge: 0, Start Time: 157.6 µs, End Time: 178.3 µs.

2. To create a link file, begin by creating a blank template. Click **+** next to the **Link** label in the **Serial Data Table** toolbar:



A **Save As** dialog will appear. Choose the **Save As Type** (.xls for Microsoft Excel format, or .csv for comma-separated text format) and then save the link file to a convenient place such as your Windows desktop.

3. Open the link file with an appropriate text or spreadsheet editor, and add number/string pairs appropriate to the target system:



The screenshot shows a spreadsheet with columns for Address, Address Description, Read/Write, Read/Write Description, Data bytes, Data bytes Description, Acknowledge, and Acknowledge Description. The data rows are: 0D, ModeReg, 02, Play, 28, Stop, A3, Pause.

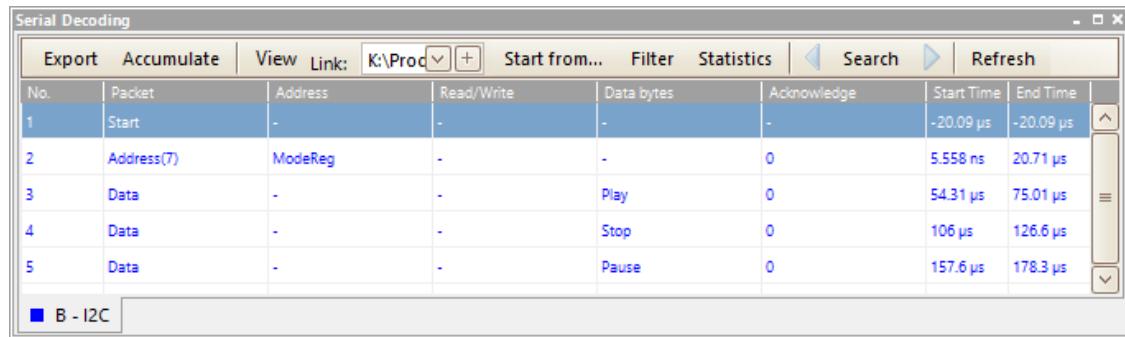
In this example we have specified that the string 'ModeReg' will appear in place of the value '0D' in the **Address** field, and so on. The numerical values are treated literally, so if the **View > Display Format** control in the data window is set to **Hex**, the link file must contain the string '0D' to match the hexadecimal value 0D. If the **View > Display Format** control is changed to **Decimal**, for example, a new link file containing decimal values will be required.

4. Save the link file.
5. Click **Browse** next to the **Link** label in the **Serial Data Table** toolbar:



6. An **Open** dialog will appear. Select the link file and click **OK**.

7. PicoScope will now replace the numeric data with the text strings that you have defined:



The screenshot shows the 'Serial Decoding' window in PicoScope 6. The window title is 'Serial Decoding'. The menu bar includes 'Export', 'Accumulate', 'View', 'Link:' (with a dropdown set to 'K:\Proc'), 'Start from...', 'Filter', 'Statistics', 'Search', and 'Refresh'. The main table displays the following data:

No.	Packet	Address	Read/Write	Data bytes	Acknowledge	Start Time	End Time
1	Start	-	-	-	-	-20.09 µs	-20.09 µs
2	Address(7)	ModeReg	-	-	0	5.558 ns	20.71 µs
3	Data	-	-	Play	0	54.31 µs	75.01 µs
4	Data	-	-	Stop	0	106 µs	126.6 µs
5	Data	-	-	Pause	0	157.6 µs	178.3 µs

At the bottom left of the window, there is a legend: a blue square followed by 'B - I2C'.

## 9 Reference

This is where you can find detailed information on the operation of PicoScope.

### 9.1 Measurement types

The [Edit Measurement dialog](#) presents a selection of measurements that PicoScope can calculate for the selected view.

#### 9.1.1 Scope measurements

**AC RMS.** The root mean square (RMS) value of the waveform *minus* the **DC Average**. It is equivalent to a *ripple* measurement.

**Cycle Time.** PicoScope will attempt to find a repeated pattern in the waveform and measure the duration of one cycle.

**DC Average.** The mean value of the waveform.

**Duty Cycle.** The amount of time that a signal spends above its mean value, expressed as a percentage of the signal period. A duty cycle of 50% means that the high time is equal to the low time.

**Falling Rate.** The rate at which the signal level falls, in signal units per second. Click the **Advanced** button in the **Add Measurement** or **Edit Measurement** dialog to specify the signal level thresholds for the measurement.

**Frequency.** The number of cycles of the waveform per second.

**Fall Time.** The time the signal takes to fall from the upper threshold to the lower threshold. Click the **Advanced** button in the **Add Measurement** or **Edit Measurement** dialog to specify the signal level thresholds for the measurement.

**High Pulse Width.** The amount of time that the signal spends above its mean value.

**Low Pulse Width.** The amount of time that the signal spends below its mean value.

**Maximum.** The highest level reached by the signal.

**Minimum.** The lowest level reached by the signal.

**Peak To Peak.** The difference between **maximum** and **minimum**.

**Rise Time.** The time the signal takes to rise from the lower threshold to the upper threshold. Click the **Advanced** button in the **Add Measurement** or **Edit Measurement** dialog to specify the signal level thresholds for the measurement.

**Rising Rate.** The rate at which the signal level rises, in signal units per second. Click the **Advanced** button in the **Add Measurement** or **Edit Measurement** dialog to specify the signal level thresholds for the measurement.

**True RMS.** The root mean square (RMS) value of the waveform, including the DC component.

**Mask Failures.** A special measurement that counts the number of failed waveforms in [Mask Limit Testing](#) mode. This measurement is added to the table automatically when you use Mask Limit Testing, so there is usually no need to select it manually.

### 9.1.2 Spectrum measurements

To add a **spectrum measurement**, open a [spectrum view](#) and then click the [Add Measurement](#) button. You can use these measurements in either [scope mode](#) or [spectrum mode](#).

**Frequency at peak.** The frequency at which the peak signal value appears.

**Amplitude at peak.** The amplitude of the peak signal value.

**Average amplitude at peak.** The amplitude of the peak signal value averaged over a number of captures.

**Total power.** The power of the whole signal captured in the spectrum view, calculated by adding the powers in all of the spectrum bins.

**Total Harmonic Distortion (THD).** The ratio of the sum of harmonic powers to the power at the fundamental frequency.

$$THD = 20 \log_{10} \left( \frac{\sqrt{V_2^2 + V_3^2 + V_4^2 + V_5^2 + V_6^2 + V_7^2}}{V_1} \right)$$

**Total Harmonic Distortion plus Noise (THD+N).** The ratio of the harmonic power plus noise to the fundamental power. THD+N values are always greater than the THD values for the same signal.

$$THD + N = 20 \log_{10} \left( \frac{\sqrt{\text{sum of squares of RMS values excluding datum}}}{\text{RMS value of datum}} \right)$$

**Spurious-free Dynamic Range (SFDR).** This is the ratio of the amplitude of the specified point (normally the peak frequency component) and the frequency component with the second largest amplitude (call it "SFDR frequency"). The component at the "SFDR frequency" is not necessarily a harmonic of the fundamental frequency component. For example, it might be a strong, independent noise signal.

**Signal+Noise+Distortion to Signal+Noise Ratio (SINAD).** The ratio, in decibels, of the signal-plus-noise-plus-distortion to noise-plus-distortion.

$$SINAD = 20 \log_{10} \left( \frac{\text{RMS value of datum}}{\sqrt{\text{sum of squares of all RMS components except datum}}} \right)$$

**Signal to Noise Ratio (SNR).** The ratio, in decibels, of the mean signal power to the mean noise power. Hanning or Blackman windows are recommended because of their low noise.

$$SNR = 20 \log_{10} \left( \frac{\text{RMS value of datum}}{\sqrt{\text{sum of squares of all values excluding datum and harmonics}}} \right)$$

**Intermodulation Distortion (IMD).** A measure of the distortion caused by the nonlinear mixing of two tones. When multiple signals are injected into a device, modulation or nonlinear mixing of these two signals can occur. For input signals at frequencies  $f_1$  and  $f_2$ , the two second-order distortion signals will be found at frequencies:  $f_3 = (f_1 + f_2)$  and  $f_4 = (f_1 - f_2)$ .

IMD is expressed as the dB ratio of the RMS sum of the distortion terms to the RMS sum of the two input tones. IMD can be measured for distortion terms of any order, but the second-order terms are most commonly used. In the second-order case, the intermodulation distortion is given by:

$$IMD = 20\log_{10} \sqrt{\frac{F_3^2 + F_4^2}{F_1^2 + F_2^2}}$$

where

$F_3$  and  $F_4$  are the amplitudes of the two second-order distortion terms (at frequencies  $f_3$  and  $f_4$  defined above)

and

$F_1$  and  $F_2$  are the amplitudes of the input tones (at frequencies  $f_1$  and  $f_2$ , as marked by the frequency rulers in the spectrum window).

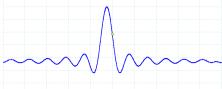
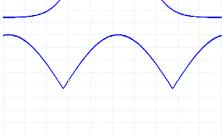
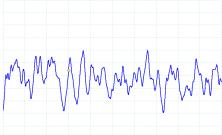
For reference, the third-order terms are at frequencies  $(2F_1 + F_2)$ ,  $(2F_1 - F_2)$ ,  $(F_1 + 2F_2)$  and  $(F_1 - 2F_2)$ .

Note: Hanning or Blackman windows are recommended because of their low noise. An FFT size of 4096 or greater is recommended in order to provide adequate spectral resolution for the IMD measurements.

**Mask Failures.** See [Mask Limit Testing](#).

## 9.2 Signal generator waveform types

The list of waveform types available in the [Signal Generator dialog](#) varies according to the type of oscilloscope connected. The full list is as follows:

Sine		Sinusoid
Square		Square wave
Triangle		Symmetrical triangle wave
RampUp		Rising sawtooth
RampDown		Falling sawtooth
Sinc		$\sin(x)/x$ , truncated on the x axis
Gaussian		The 'bell curve' of the normal distribution, truncated on the x axis
HalfSine		A rectified sinusoid
WhiteNoise		Random samples at the maximum update rate of the AWG
PRBS		Pseudo-random binary sequence - a random sequence of bits with adjustable bit rate
DCVoltage		Constant voltage, adjustable using the <b>Offset</b> control
Arbitrary		Any waveform created by the arbitrary waveform editor

### 9.3 Spectrum window functions

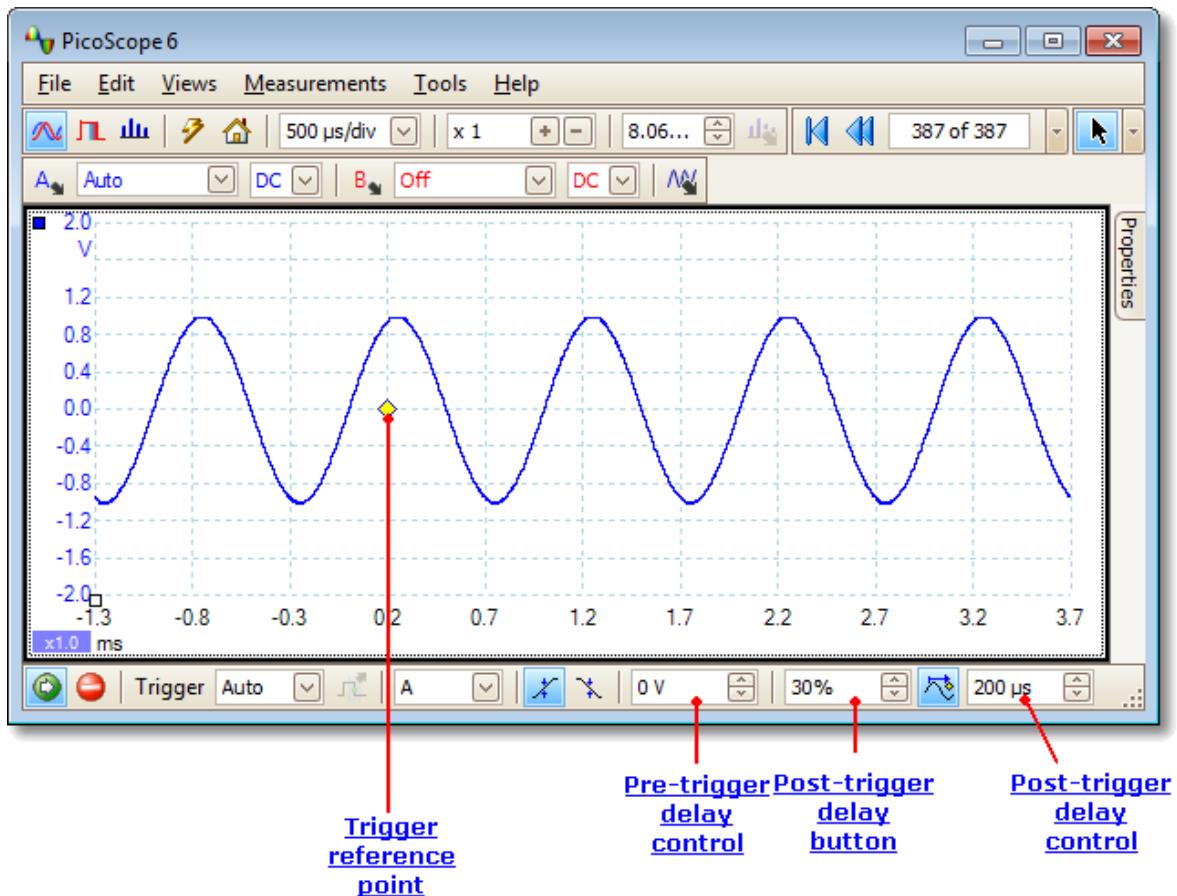
To create a [spectrum view](#), PicoScope captures a block of sampled data over a finite time interval and then uses a Fast Fourier Transform to compute its spectrum. The algorithm assumes a signal level of zero at all times outside the captured time interval. Typically, this assumption causes sharp transitions to zero at either end of the data, and these transitions have an effect on the computed spectrum, creating unwanted artefacts such as ripple and gain errors. To reduce these artefacts, the signal can be faded in and out at the start and end of the block. There are several commonly used "window functions" that can be convolved with the data to effect this fading, and which are chosen according to the type of signal and the purpose of the measurement.

The **Window Functions control** in the [Spectrum Options dialog](#) lets you select one of the standard window functions for spectrum analysis. The following table shows some of the figures of merit used to compare the functions.

Window	Main peak width (bins @ -3 dB)	Highest side lobe (dB)	Side lobe roll-off (dB/octave)	Notes
<b>Blackman</b>	1.68	-58	18	often used for audio work
<b>Gaussian</b>	1.33 to 1.79	-42 to -69	6	gives minimal time and frequency errors
<b>Triangular</b>	1.28	-27	12	also called Bartlett window
<b>Hamming</b>	1.30	-41.9	6	also called raised sine-squared; used in speech analysis
<b>Hann</b>	1.20 to 1.86	-23 to -47	12 to 30	also called sine-squared; used for audio & vibration
<b>Blackman-Harris</b>	1.90	-92	6	general-purpose
<b>Flat-top</b>	2.94	-44	6	negligible pass-band ripple; used mainly for calibration
<b>Rectangular</b>	0.89	-13.2	6	no fading; maximal sharpness; used for short transients

## 9.4 Trigger timing (part 1)

The **pre-trigger time control** and **post-trigger delay control** functions are described individually under "[Triggering toolbar](#)", but the interaction between the two controls is also important to understand. Here is a screen shot of a [scope view](#) with post-trigger delay enabled:



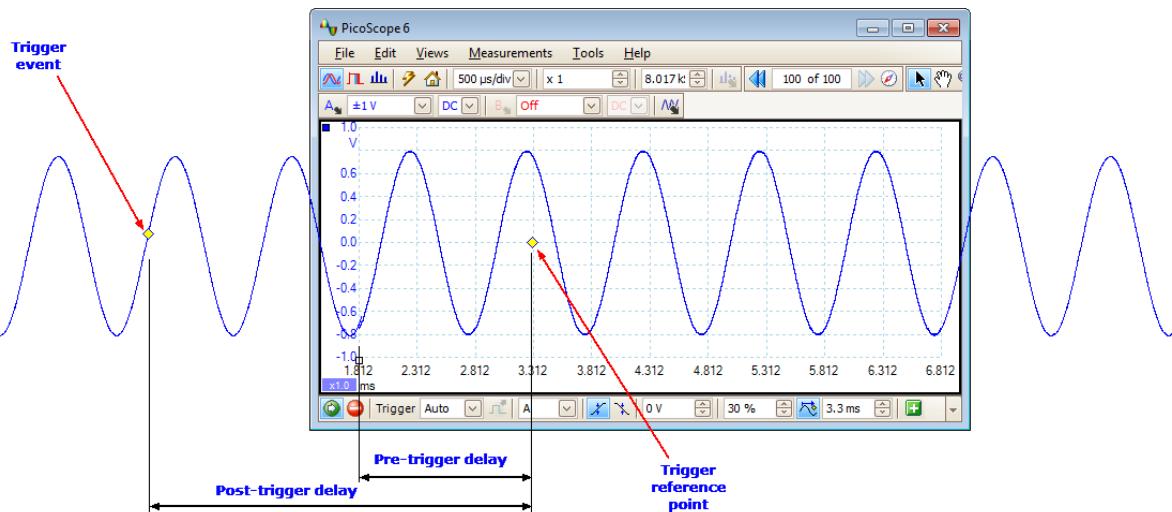
Note 1. The trigger reference point (◊) does not lie on the waveform. This is because the post-trigger delay is set to 200 µs, which means that the trigger occurred 200 µs before the reference point, somewhere off the left-hand edge of the [scope view](#). The time axis is aligned so that the trigger reference point is at 200 µs.

Note 2. The pre-trigger delay is set to 25%, which makes the trigger reference point appear 25% of the way across the scope view from the left-hand edge.

Note 3. PicoScope limits the trigger-to-reference-point delay to a multiple of the total capture time. Once you have reached this limit, the program will not let you increase the pre-trigger delay, and if you increase the post-trigger delay, PicoScope will reduce the pre-trigger delay to stop the total exceeding the limit. The multiple is typically 100 in most trigger modes, and 1 in [ETS](#) mode.

## 9.5 Trigger timing (part 2)

"[Trigger timing \(part 1\)](#)" introduced the concepts of [pre-trigger delay](#) and the [post-trigger delay](#). This diagram below shows how they are related.



The **pre-trigger delay** positions the [scope view](#) in relation to the trigger reference point so that you can choose how much of the waveform should be before the reference point, and how much after it.

The **post-trigger delay** is like the delayed trigger of a conventional oscilloscope. PicoScope waits for this time after the trigger event before drawing the trigger reference point. Scope devices have a limit to the number of sampling intervals that can elapse between the trigger event and the end of the capture, so the software may adjust the pre-trigger delay to keep within this limit.

**Tip:** If you have set up a post-trigger delay, you can click the post-trigger delay button while the scope is running whenever you want to switch between viewing the trigger event and the trigger reference point.

## 9.6 Serial protocols

The [serial decoding](#) feature of PicoScope understands the following serial protocols.

### 9.6.1 RS232/UART protocol

You can decode **RS232 (UART)** data using the [serial decoding](#) feature built into PicoScope.

#### About RS232

RS232 is the serial data standard used by UARTs (Universal Asynchronous Receiver/Transmitters) in the "serial" or "COM" ports once commonly found on computers. It was developed in the 1960s for connecting modems to terminals. The full standard uses a voltage swing of  $\pm 12$  V, larger than most other standards. The simplest RS232 connection consists of two signals: Rx (receive) and Tx (transmit).

The in-window view of the data looks like this:

Serial Decoding						
No.	Packet	Start Bit	Data bytes	Parity Bit	Stop Bit	
6472	Data	0	5F	-	1	
5355	Data	0	5F	-	1	
1070	Data	0	5F	-	1	
9155	Data	0	5F	-	1	

**A - RS232/UART**

The columns in the data table are as follows:

Column No.	Description
No.	Serial number of the packet (decimal). If <b>Accumulate</b> mode is off, this counts from the start of the selected waveform. If <b>Accumulate</b> mode is on, this counts from the start of the first waveform in the waveform buffer.
Packet	Packet type: all packets in this format are classified as <b>Data</b> .
Start Bit	If present, this is the fixed "1" bit at the start of the word.
Data bytes	Contents of data packets.
Parity Bit	The error-correction bit, if present, at the end of the word.
Stop Bit	If present, this is the fixed "1" bit at the end of the word.
Error	Indicates whether there was a data error.
Start Time	Time according to PicoScope timebase at start of frame.
End Time	Time according to PicoScope timebase at end of frame.
Packet Time*	The duration of the packet (End Time – Start Time).
Min Voltage*	Minimum voltage.
Max Voltage*	Maximum voltage.
Voltage Delta*	Voltage range of the signal (Max Voltage – Min Voltage).

\* Display of these items is toggled on and off by the **Statistics** button.

## 9.6.2 SPI protocol

You can decode **SPI Bus** data using the [serial decoding](#) feature built into PicoScope.

### About SPI Bus

SPI (Serial Peripheral Interface) Bus is a serial data standard used for communication between microprocessors and peripheral devices. It was developed by Motorola. The original standard uses a 4-wire link, although 3-wire and 2-wire versions are also used.

The in-window view of the data looks like this:

Serial Decoding				
Export   Accumulate   View   Link: <None>			Start from...	Filter   Statistics
No.	Packet	Data bytes	Start Time	End Time
1	Unknown	-	791.2 µs	891.7 µs
2	Data	80	1.799 ms	1.888 ms
3	Data	F8	3.091 ms	3.18 ms
4	Data	80	4.605 ms	4.693 ms
5	Data	00	5.696 ms	5.985 ms

■ A - SPI

The columns in the data table are as follows:

Column No.	Description
No.	Serial number of the packet (decimal). If <b>Accumulate</b> mode is off, this counts from the start of the selected waveform. If <b>Accumulate</b> mode is on, this counts from the start of the first waveform in the waveform buffer.
Packet	Packet type: <b>Start</b> , <b>Stop</b> , <b>Address</b> , <b>Data</b> or <b>Unknown</b> .
Data bytes	Contents of data packets.
Start Time	Time according to PicoScope timebase at start of frame.
End Time	Time according to PicoScope timebase at end of frame.
Packet Time*	The duration of the packet (End Time – Start Time).
Min Voltage*	Minimum voltage.
Max Voltage*	Maximum voltage.
Voltage Delta*	Voltage range of the signal (Max Voltage – Min Voltage).

\* Display of these items is toggled on and off by the **Statistics** button.

### 9.6.3 I<sup>2</sup>C protocol

You can decode **I<sup>2</sup>C Bus** data using the [serial decoding](#) feature built into PicoScope.

#### About I<sup>2</sup>C Bus

I<sup>2</sup>C (Inter-Integrated Circuit) Bus is a serial protocol used mainly in consumer electronics for communications between devices on the same circuit board, and between computers and displays. The standard was originally developed in the 1980s by Philips. It uses two signals: clock (SCL) and data (SDA).

The in-window view of the data looks like this:

Serial Decoding														
Export		Accumulate		View		Link:		None	Start from...	Filter	Statistics	Search	Refresh	Clear
No	Packet	Address	Read/Write	Data bytes			Acknowledge	Start Time	End Time					
2	Address	70	WRITE	-	-	-	Yes	791.3 µs	891.7 µs					
3	Data	-	-	01	-	-	Yes	1.799 ms	1.9 ms					
4	Data	-	-	02	-	-	Yes	1.91 ms	2.01 ms					
7	Address	70	WRITE	-	-	-	Yes	2.083 ms	2.183 ms					
8	Data	-	-	1F	-	-	Yes	3.091 ms	3.191 ms					
9	Data	-	-	08	-	-	Yes	3.202 ms	3.302 ms					
10	Data	-	-	C0	-	-	Yes	3.313 ms	3.413 ms					
11	Data	-	-	A7	-	-	Yes	3.424 ms	3.524 ms					
14	Address	70	WRITE	-	-	-	Yes	3.597 ms	3.698 ms					

**■ A - I<sup>2</sup>C Data**

The columns in the data table are as follows:

Column No.	Description
<b>No.</b>	Serial number of the packet (decimal). If <b>Accumulate</b> mode is off, this counts from the start of the selected waveform. If <b>Accumulate</b> mode is on, this counts from the start of the first waveform in the waveform buffer.
<b>Packet</b>	Packet type: <b>Start</b> , <b>Stop</b> , <b>Address</b> , <b>Data</b> or <b>Unknown</b> .
<b>Address</b>	Shown for address packets.
<b>Read/Write</b>	Polarity of the Read/Write flag.
<b>Data bytes</b>	Contents of data packets.
<b>Acknowledge</b>	Whether the destination acknowledged the packet.
<b>Baud Rate*</b>	The signaling rate detected for this packet.
<b>Start Time</b>	Time according to PicoScope timebase at start of frame.
<b>End Time</b>	Time according to PicoScope timebase at end of frame.
<b>Packet Time*</b>	The duration of the packet (End Time – Start Time).
<b>Min Voltage*</b>	Minimum voltage.
<b>Max Voltage*</b>	Maximum voltage.
<b>Voltage Delta*</b>	Voltage range of the signal (Max Voltage – Min Voltage).

\* Display of these items is toggled on and off by the **Statistics** button.

### 9.6.4 I<sup>2</sup>S protocol

You can decode **I<sup>2</sup>S Bus** data using the [serial decoding](#) feature built into PicoScope.

#### About I<sup>2</sup>S Bus

I<sup>2</sup>S (Inter-IC Sound) Bus is a serial protocol used in digital audio devices for communications between circuits such as CD transports and audio DACs. The standard was originally developed in by Philips Semiconductors. It uses three signals: clock (SCK), word select (WS) and data (SD).

The in-window view of the data looks like this:

Serial Decoding				
No.	Packet	Data bytes	Start Time	End Time
1	Data	0	175 ns	178.1 ns
2	Data	0	375 ns	378.1 ns
3	Data	0	1.175 $\mu$ s	1.178 $\mu$ s
4	Data	1	4.825 $\mu$ s	4.828 $\mu$ s
5	Data	1	7.775 $\mu$ s	7.778 $\mu$ s
6	Data	0	8.575 $\mu$ s	8.578 $\mu$ s
7	Data	0	9.025 $\mu$ s	9.028 $\mu$ s
8	Data	1	10.33 $\mu$ s	10.33 $\mu$ s
9	Data	0	10.68 $\mu$ s	10.68 $\mu$ s

A - I<sup>2</sup>S Data(BETA)

The columns in the data table are as follows:

Column	Description
<b>No.</b>	Serial number of the packet (decimal). If <b>Accumulate</b> mode is off, this counts from the start of the selected waveform. If <b>Accumulate</b> mode is on, this counts from the start of the first waveform in the waveform buffer.
<b>Packet</b>	Packet type: all I <sup>2</sup> S packets are of type <b>Data</b> .
<b>Data bytes</b>	Contents of data packets.
<b>Baud Rate*</b>	The signaling rate detected for this packet.
<b>Start Time</b>	Time according to PicoScope timebase at start of frame.
<b>End Time</b>	Time according to PicoScope timebase at end of frame.
<b>Packet Time*</b>	The duration of the packet (End Time – Start Time).
<b>Min Voltage*</b>	Minimum voltage.
<b>Max Voltage*</b>	Maximum voltage.
<b>Voltage Delta*</b>	Voltage range of the signal (Max Voltage – Min Voltage).

\* Display of these items is toggled on and off by the **Statistics** button.

### 9.6.5 CAN protocol

You can decode **CAN Bus** data using the [serial decoding](#) feature built into PicoScope.

## About CAN Bus

CAN (Controller Area Network) Bus is a serial protocol used in automotive and industrial machinery to allow microcontrollers to communicate with each other. The standard was originally developed in 1983 by Robert Bosch GmbH. It typically uses differential signalling (with signals named CAN H and CAN L) to increase noise immunity.

The in-window view of the data looks like this:

Serial Decoding								
Export		Accumulate		View		Link: <None>		
No.	Packet	Address	Read/Write	Data bytes		Acknowledge	Start Time	End Time
2	Address	70	WRITE	-	-	Yes	4.215 ms	4.316 ms
3	Data	-	-	01	-	Yes	5.222 ms	5.323 ms
4	Data	-	-	02	-	Yes	5.334 ms	5.434 ms
7	Address	70	WRITE	-	-	Yes	5.507 ms	5.607 ms
8	Data	-	-	1F	-	Yes	6.514 ms	6.615 ms

**A - I2C**

The columns in the data table are as follows:

Column No.	Description
No.	Serial number of the frame (decimal). If <b>Accumulate</b> mode is off, this counts from the start of the selected waveform. If Accumulate mode is on, this counts from the start of the first waveform in the waveform buffer.
ID	Identifier (hexadecimal). Should be unique for a specified data type. Base frames have an 11-bit identifier, and extended frames have a 29-bit identifier.
Frame	CAN Bus data is divided into frames, each consisting of a number of bits. A frame can be any of the following types:  <b>Data:</b> contains data destined for a node <b>Remote:</b> a request for the transmission of a specific identifier <b>Error:</b> transmitted by a node that detects an error <b>Overload:</b> inserted to add a delay between frames <b>Interframe:</b> time interval preceding data frames and remote frames  Any frame can also be a base frame or an extended frame.
RTR	Remote transmission request
SRR	Used only in extended frames
IDE	Identifier extension bit
R0	Reserved bit
R1	Reserved bit, extended frames only
DLC	Data length code. Indicates the number of bytes of data.
Data bytes	The data content of the message, <b>DLC</b> bytes long (hexadecimal)

<b>CRC Sequence</b>	A cyclic redundancy check of the data (hexadecimal)
<b>CRC Delimiter</b>	A fixed bit following the <b>CRC Sequence</b> field
<b>Ack Slot</b>	Node asserts this bit to acknowledge receipt
<b>Ack Delimiter</b>	A fixed bit following the <b>Ack Slot</b> field
<b>Error</b>	Set if PicoScope detects an error
<b>Stuffed Bits*</b>	The number of extra bits inserted in the frame to assist synchronization
<b>Baud Rate*</b>	Number of bits per second.
<b>Start Time</b>	Value of PicoScope timebase at start of frame.
<b>End Time</b>	Value of PicoScope timebase at end of frame.
<b>Frame Time*</b>	Duration of frame (End Time – Start Time).
<b>Min Voltage*</b>	Minimum voltage.
<b>Max Voltage*</b>	Maximum voltage.
<b>Voltage Delta*</b>	Voltage range of the signal (Max Voltage – Min Voltage).

\* Display of these items is toggled on and off by the **Statistics** button.

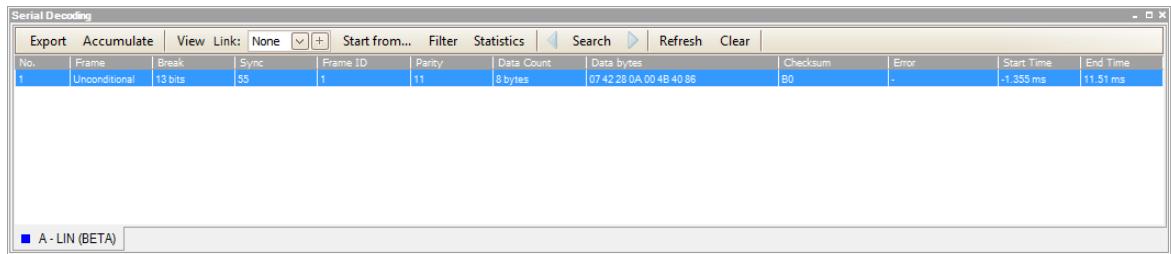
### 9.6.6 LIN protocol

You can decode **LIN** data using the [serial decoding](#) feature built into PicoScope.

#### About LIN

LIN (Local Interconnect Network) is a serial protocol used in automotive electronics to allow microcontrollers to communicate with low-speed peripherals. The standard was defined by the LIN Consortium, a group of five vehicle manufacturers. It uses a single wire pair with a master-slave topology.

The in-window view of the data looks like this:



The columns in the data table are as follows:

Column No.	Description
No.	Serial number of the frame (decimal). If <b>Accumulate</b> mode is off, this counts from the start of the selected waveform. If <b>Accumulate</b> mode is on, this counts from the start of the first waveform in the waveform buffer.
Frame	LIN Bus data is divided into frames, each consisting of a number of bits. A frame can be any of the following types:
	<b>Unconditional:</b> all subscribers must receive this frame.
	<b>Event-triggered:</b> multiple slaves may respond to this, but only if they have new data.
	<b>Sporadic:</b> transmitted by the master when it knows that the slave contains updated data.
	<b>Diagnostic:</b> contains diagnostic or configuration data.
	<b>User-Defined:</b> depends on your application.
	<b>Reserved:</b> should not be used.
Break	The start-of-frame notifier.
Sync	A fixed value (0x55) used for automatic baud rate detection.
Frame ID	A six-bit value to identify the type of data carried by the frame.
Parity	A two-bit value calculated on the data.
Data count	Number of bytes of data in the frame.
Data bytes	The data content of the message.

<b>Checksum</b>	A cyclic redundancy check of the data (hexadecimal).
<b>Calculated Checksum*</b>	The checksum expected by PicoScope.
<b>Error</b>	Set if PicoScope detects an error.
<b>Start Time</b>	Value of PicoScope timebase at start of frame.
<b>End Time</b>	Value of PicoScope timebase at end of frame.
<b>Packet Time*</b>	The duration of the packet (End Time – Start Time).
<b>Min Voltage*</b>	Minimum voltage.
<b>Max Voltage*</b>	Maximum voltage.
<b>Voltage Delta*</b>	Voltage range of the signal (Max Voltage – Min Voltage).

\* Display of these items is toggled on and off by the **Statistics** button.

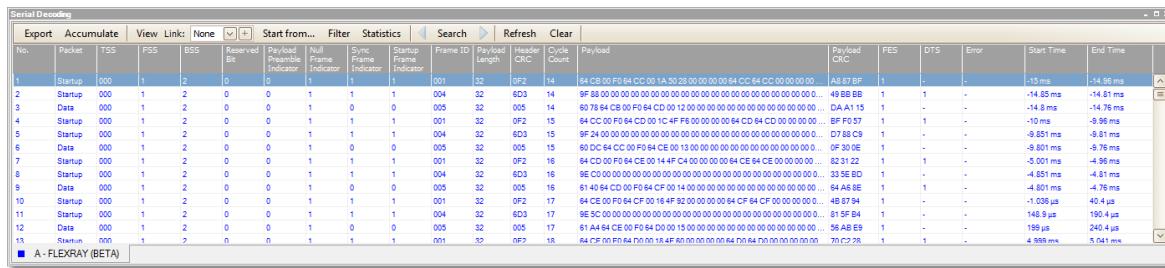
### 9.6.7 FlexRay protocol

You can decode **FlexRay** data using the [serial decoding](#) feature built into PicoScope.

#### About FlexRay

FlexRay is an automotive network communications protocol for high-speed data. It is designed to be fault-tolerant.

The in-window view of the data looks like this:



The columns in the data table are as follows:

#### Column No.

#### Description

Serial number of the packet (decimal). If **Accumulate** mode is off, this counts from the start of the selected waveform. If **Accumulate** mode is on, this counts from the start of the first waveform in the waveform buffer.

#### Packet

The type of packet.

#### TSS

Transmission Start Signal. Bit 0 of the frame.

#### FSS

Frame Start Signal. Bit 1 of the frame.

#### BSS

Byte Start Signal. A fixed 2-bit code.

#### Reserved Bit

Not used.

#### Payload Preamble Indicator

Indicates the presence of a network management vector or message ID.

#### Null Frame Indicator

Indicates that the payload contains no useful payload data.

#### Sync Frame Indicator

Indicates whether the frame is a synchronization frame.

#### Frame ID

Defines the slot in which the frame is transmitted.

#### Payload Length

The number of payload bytes divided by 2.

#### Header CRC

Cyclic redundancy check of data from the sync frame indicator to the payload length.

#### Cycle Count

The value of the cycle counter.

#### Payload

0 to 254 bytes of data.

<b>Payload CRC</b>	CRC for the frame.
<b>FES</b>	Frame End Signal.
<b>DTS</b>	Dynamic Trailing Sequence.
<b>Error</b>	Indicates whether an error was detected.
<b>Start Time</b>	Time according to PicoScope timebase at start of frame.
<b>End Time</b>	Time according to PicoScope timebase at end of frame.
<b>Packet Time*</b>	The duration of the packet (End Time – Start Time).
<b>Min Voltage*</b>	Minimum voltage.
<b>Max Voltage*</b>	Maximum voltage.
<b>Voltage Delta*</b>	Voltage range of the signal (Max Voltage – Min Voltage).

\* Display of these items is toggled on and off by the **Statistics** button.

## 9.7 Device feature table

Some PicoScope 6 features require special hardware and are therefore not available on all devices. Feature availability is indicated in the table below. Please see the relevant device data sheet for more details.

Series / Model	50	AT	AW	AX	BL	DC	DI	EX	FC	FR	LP	RA	RU	SG	ST	SW
<b>ADC-212*</b>											✓					
<b>USB DrDAQ</b>								✓								✓
<b>PicoLog 1000</b>																
<b>PicoScope 2202-2203</b>																
<b>PicoScope 2204-2205</b>	✓	✓													✓	✓
<b>PicoScope 2206-2208</b>	✓	✓							✓			✓		✓	✓	✓
<b>PicoScope 2204A</b>	✓	✓												✓		✓
<b>PicoScope 2205A</b>	✓	✓												✓		✓
<b>PicoScope 2206A</b>																
<b>PicoScope 2207A</b>	✓	✓						✓					✓	✓	✓	✓
<b>PicoScope 2208A</b>																
<b>PicoScope 2000 MSO</b>	✓	✓						✓					✓	✓	✓	✓
<b>PicoScope 3204</b>	✓											✓				✓
<b>PicoScope 3205</b>	✓											✓		✓		✓
<b>PicoScope 3206</b>	✓													✓		✓
<b>PicoScope 3000A/B</b>	✓	✓						✓		✓			✓	✓	✓	✓
<b>PicoScope 3000 MSO</b>	✓	✓						✓	✓				✓	✓	✓	✓
<b>PicoScope 3000D MSO</b>																
<b>PicoScope 4223-4224</b>	✓										✓		✓	✓	✓	
<b>PicoScope 4423-4424</b>																
<b>PicoScope 4225/4425</b>	✓									✓		✓	✓	✓		
<b>PicoScope 4226-4227</b>	✓	✓							✓			✓	✓	✓	✓	✓
<b>PicoScope 4262</b>	✓	✓						✓		✓		✓	✓	✓	✓	✓
<b>PicoScope 4824</b>	✓	✓						✓				✓	✓	✓	✓	✓
<b>PicoScope 5000</b>	✓	✓								✓		✓	✓	✓	✓	✓
<b>PicoScope 5000A/B</b>	✓	✓						✓	✓	✓		✓	✓	✓	✓	✓
<b>PicoScope 6000</b>	✓	✓	✓	✓	✓	✓	✓					✓	✓	✓	✓	✓

\* Automotive software only

50	<a href="#">50 ohm inputs</a>
AT	<a href="#">Advanced triggers</a>
AW	<a href="#">Arbitrary waveform generator</a>
AX	<a href="#">Auxiliary input/output</a>
BL	<a href="#">Switchable bandwidth limiter</a>
DC	<a href="#">DC offset adjustment</a>
DI	<a href="#">Digital inputs</a>
EX	<a href="#">External trigger input</a>
FC	<a href="#">Frequency counter</a>
FR	<a href="#">Flexible resolution</a>
LP	<a href="#">Lowpass filtering</a>
RA	<a href="#">Rapid triggering</a>
RU	<a href="#">Runt pulse triggering</a>
SG	<a href="#">Signal generator</a>
ST	<a href="#">Signal generator triggers</a>
SW	<a href="#">Signal generator sweep mode</a>

## 9.8 Command-line syntax

PicoScope can be run from the Windows command line, allowing you to run tasks manually or under the control of a batch file or other program.

### To display the GUI

`PicoScope <filename>`

`<filename>` Specifies a single `.psdata` or `.pssettings` file.

Example: `PicoScope C:\Temp\source.psdata`

### To display help

`PicoScope /?`

Shows help on all command-line options.

### To convert a psdata file

`PicoScope /C,/c`

Converts a psdata file from one format to another. Cannot be used with `/p[rint]`.

Syntax:

`PicoScope /c[onvert] <names> [/d <names>] /f <format> [/q] [/b [<n>[:<m>]] | [all]] [/v <viewportname>]`

`<names>` Specifies a list of one or more directories or psdata files. Wildcards may be used to specify multiple files. If a directory is specified, all psdata files within that directory will be specified. This is a mandatory argument.

`/d <names>` Destination. Default is input filename with new extension.

`/f <format>` Destination format: csv, txt, png, bmp, gif, agif [animated GIF], psdata, pssettings, mat [MATLAB]. This is a mandatory argument.

`/q` Quiet mode. Do not ask before overwriting files. The default is to prompt.

`/b [<n>[:<m>]] | all` Waveform number n, waveform range n to m or all waveforms. The default is the current waveform.

`/v <viewportname>` View to convert. Default is current view.

Example:

`PicoScope /c C:\Temp\source.psdata /f png /b 5:9 /v Scope2`

## To print a view

`PicoScope /P, /p`

Prints a view in the psdata file. Cannot be used with `/c[onvert]`.

Syntax:

`PicoScope /p[rint] <names> [/b [<n>[:<m>]] | all] [/v <viewportname>]`

`<names>` Specifies a list of one or more directories or .psdata files. Wildcards may be used to specify multiple files. If a directory is specified, all .psdata files within that directory will be specified. This is a mandatory argument.

`/b [<n>[:<m>]] | all` Waveform number n, waveform range n to m or all buffers. Default is current waveform.

`/v <viewportname>` View to convert. Default is current view.

Example:

`PicoScope /p C:\Temp\source.psdata /b 5:9 /v Scope2`

## To import notes

`PicoScope /N, /n`

Copies text from a specified file to the [Notes area](#).

Syntax:

`PicoScope /n[otes] <notes filename> <filename>`

`<notes filename>` Specifies a single text file.

`<filename>` Specifies a single psdata or psettings file.

Example:

`PicoScope /n C:\Temp\source.txt C:\Temp\source.psdata`

## To run an automation command

`PicoScope /A, /a`

Run automation command or macro on an existing instance of PicoScope 6.

Syntax:

`PicoScope /a[utomation] <command> | <macro>`

`<command>` Automation command

`<macro>` Path to `.psmacro` file containing a [macro](#)

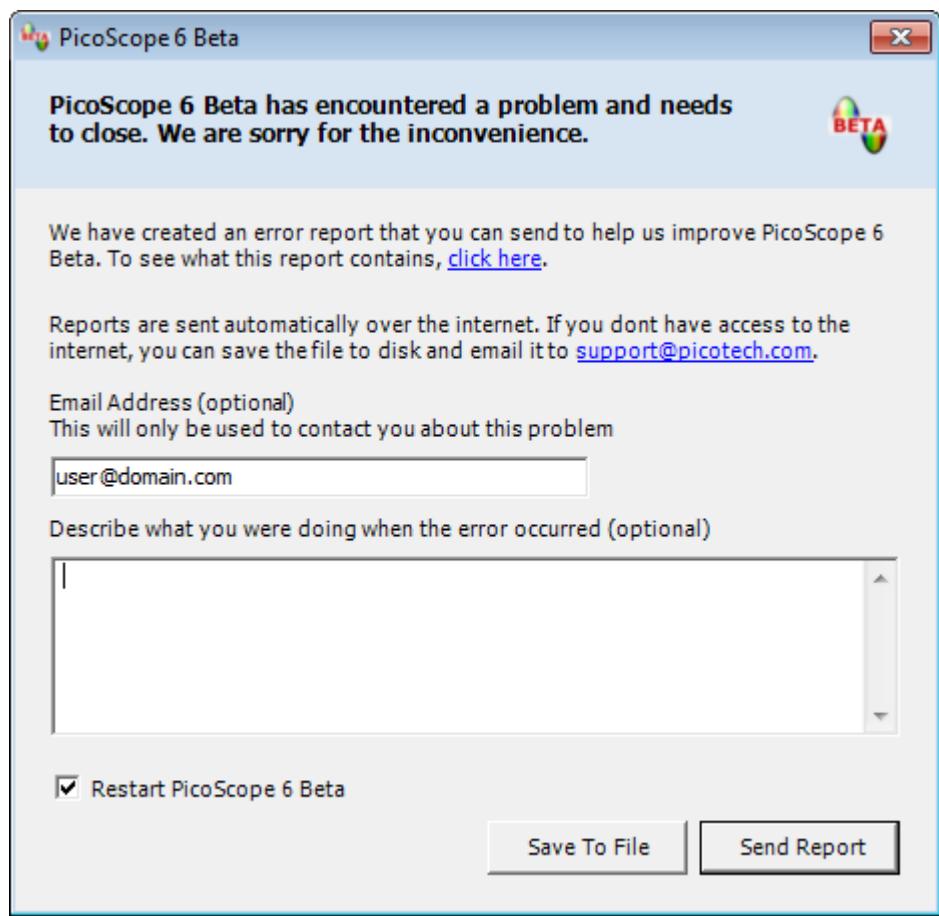
**Examples:**

```
PicoScope /a Run.Pressed=True
PicoScope /a MyMacro.psmacro
PicoScope /a ? (prints a list of automation commands to the console)
PicoScope /a Measurements? (prints information on the Measurements command)
```

An instance of PicoScope must be running before any automation command is issued.

## 9.9 Application Error dialog

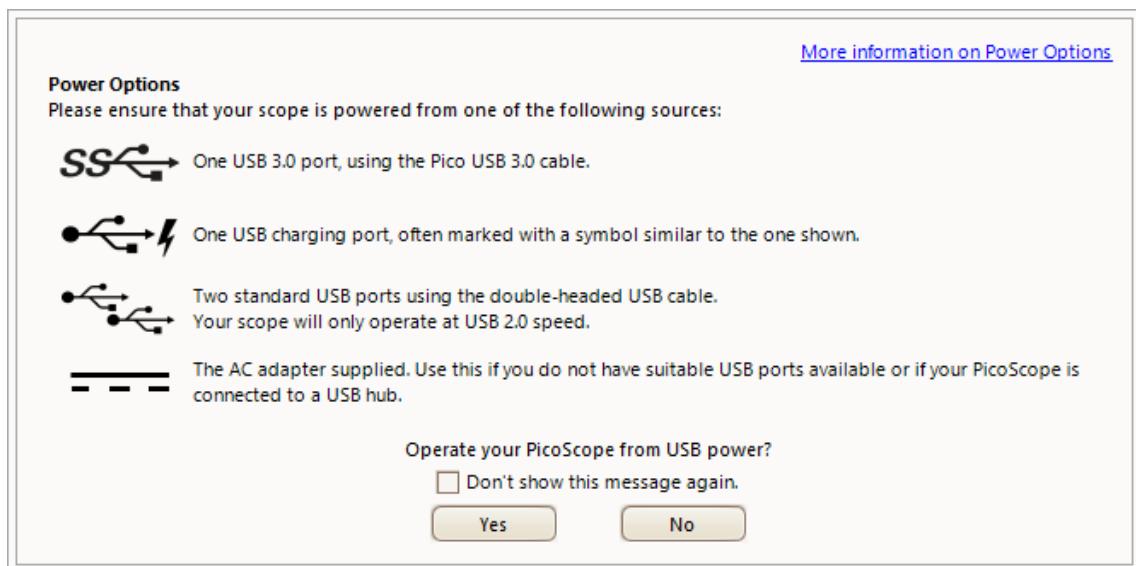
If there is a problem with PicoScope and the program needs to close, the **Application Error dialog** appears:



It would be very helpful to us if you could send us a report of the problem. All you need to do is click the **Send Report** button, then save the **.zip** file in a place where you can easily find it again, such as on your desktop. Then, email the **.zip** file to us at [support@picotech.com](mailto:support@picotech.com) and we will do the rest. Don't worry if you can't think of a comment to enter in the text box: just leave it empty and send the report to us anyway.

## 9.10 Flexible power

The flexible power system for PicoScope devices provides a choice of power sources. In most cases a straightforward USB connection will be enough to power the scope. If PicoScope needs you to switch to a different powering method, it will display a dialog similar to this:



### The power icons explained



**USB 3.0 port.** If this symbol appears, you may power the scope from any USB 3.0 port using the USB 3.0 cable provided with your scope.



**USB charging port.** If this symbol appears, you may power the scope from any USB port capable of supplying 1200 mA of current.



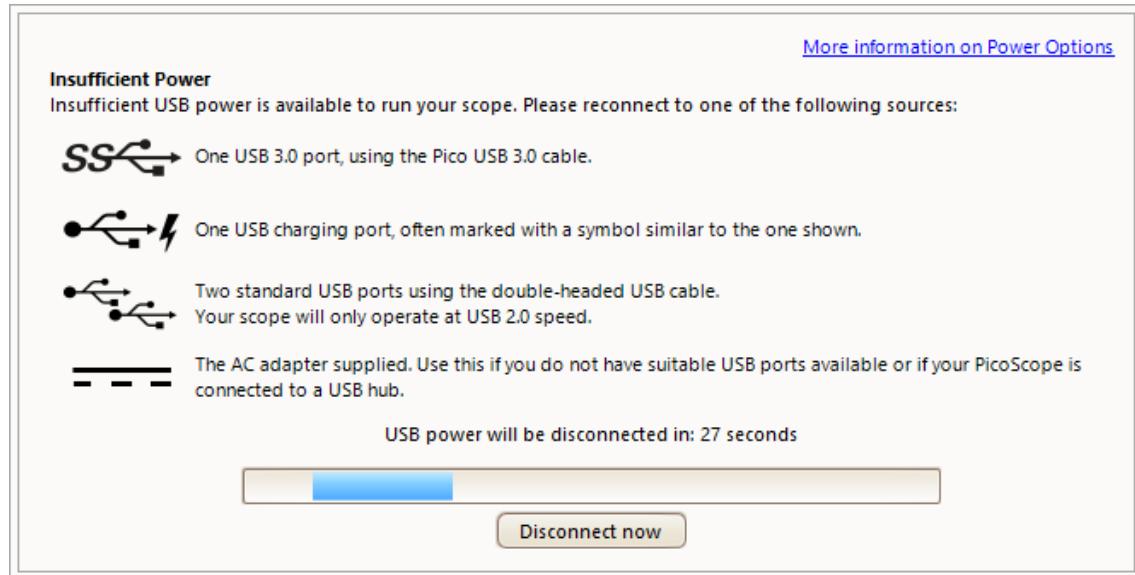
**Dual USB port power.** If this symbol appears, you may power the scope from any two powered USB ports using the dual-headed USB 2.0 cable supplied with your scope.



**AC adapter.** Use the adapter supplied and plug it into the "DC IN" socket on your scope. If you power the scope in this way, the **Power Options** dialog will automatically close and the scope will start working immediately.

## Insufficient USB power

If insufficient USB power is available, PicoScope displays this dialog:



Connect one of the power sources listed and try again. If you plug in the AC adapter, the dialog will automatically close and you can continue working.

## 9.11 Glossary

**AC coupling.** In this mode, the scope device rejects very low signal frequencies below about 1 hertz. This allows you to use the full resolution of the scope to measure a.c. signals accurately, ignoring any DC offset. You cannot measure the signal level with respect to ground in this mode.

**AWG.** An arbitrary waveform generator (AWG) is a circuit that can generate a waveform of almost any shape. It is programmed with a data file, supplied by the user, which defines the output voltage at a number of equally spaced points in time. The circuit uses this data to reconstruct the waveform with a specified amplitude and frequency.

**Axis.** A line marked with measurements. PicoScope shows one vertical axis for each channel that is enabled in a view, giving measurements in volts or other units. Each view also has a single horizontal axis, which is marked in units of time for a scope view, or units of frequency for a spectrum view.

**Channel.** A scope device has one or more channels, each of which can sample one signal. High-speed scope devices typically have one BNC connector per channel.

**CSV.** Comma-separated values. A text file containing tabulated data, with columns separated by commas and rows by line-breaks. CSV format is used for importing and exporting PicoScope [arbitrary waveform files](#). You can also export PicoScope waveforms in CSV format. CSV files can be imported into spreadsheets and other programs.

**DC coupling.** In this mode, the scope device measures the signal level relative to signal ground. This shows both DC and AC components.

**Dead time.** The time between the end of one capture and the start of the next. To obtain the minimum possible dead time, use **Rapid** trigger mode.

**Demo mode.** If PicoScope is started when no scope device is plugged in, it allows you to select a "demo device", a virtual scope unit that you can use to test the software. The program is then in "demo" (short for "demonstration") mode. This mode provides a simulated, configurable signal source for each input channel of the demo device.

**ETS.** Equivalent Time Sampling. A method of increasing the effective sampling rate of the scope. In a scope view, the program captures several cycles of a repetitive signal, then combines the results to produce a single waveform with higher time-resolution than a single capture. For accurate results, the signal must be perfectly repetitive and the trigger must be stable.

**Floating inputs.** A feature of the PicoScope 4225 and 4425 oscilloscopes. These are inputs that do not share a common measurement ground. The channels are separated by high impedances and their measurement grounds can be connected to any voltage within the specification range. It is important however that every input in use has both a signal and a ground connection.

**Graticule.** The horizontal and vertical dashed lines in every view. These help you estimate the amplitude and time or frequency of features on the waveform.

**Grid.** The arrangement of viewports. The number of grid rows and the number of grid columns can each be either 1, 2, 3 or 4.

**IEPE.** Integrated circuit piezoelectric. A type of sensor, usually built to detect acceleration, vibration or sound, and with a built-in amplifier. IEPE sensors can only be used with special PicoScope oscilloscopes that have IEPE-compatible inputs.

**In focus.** PicoScope can display several views, but only one view is in focus at any time. When you click a toolbar button, it will usually affect only the view that is in focus. To bring a view into focus, click on it.

**MSO.** Mixed-signal oscilloscope. An instrument that captures and displays analog and digital signals on the same timebase.

**PC Data Logger.** A measuring instrument consisting of a hardware interface and the PicoLog software running on a PC. You can also use the device with the PicoScope software to create a multi-channel voltage input oscilloscope.

**PC Oscilloscope.** A measuring instrument consisting of a [scope device](#) and the PicoScope software running on a PC. A PC Oscilloscope has the same functions as a traditional bench-top oscilloscope but is more flexible and cost-effective. You can improve its performance by upgrading the PC using standard parts from any computer shop, or by purchasing a new scope device, and you can upgrade the software by downloading an update from Pico Technology.

**Probe.** An accessory that attaches to your oscilloscope and picks up a signal to be measured. Probes are available to pick up any form of signal, but they always deliver a voltage signal to the oscilloscope. PicoScope has built-in definitions of standard probes, but also allows you to define custom probes.

**Progressive mode.** Normally, PicoScope redraws the waveform in a scope view many times every second. At timebases slower than 200 ms/div, however, it switches to progressive mode. In this mode, PicoScope updates the scope view continuously as each capture progresses, rather than waiting for a complete capture before updating the view.

**Resolution enhancement.** Collecting samples at a faster rate than requested, then combining the excess samples by averaging. This technique can increase the effective resolution of a scope device when there is a small amount of noise on the signal. [\(More details.\)](#)

**Ruler.** A vertical or horizontal dashed line that can be dragged into place on a waveform in a view. PicoScope displays the signal level, time value or frequency value of all rulers in the Ruler Legend box.

**Scope device.** The box from Pico Technology that you plug into the USB or parallel port of your computer. With the help of the PicoScope software, the scope device turns your computer into a PC Oscilloscope.

**Standard deviation.** A statistical measure of the spread of a set of samples. The standard deviation of the set  $y_0 \dots y_{n-1}$  is defined as:

$$SD = \sqrt{\sum_{i=0}^{n-1} (y_i - \bar{y})^2},$$

where  $\bar{y}$  is the arithmetic mean of all the samples. The units of the standard deviation value are the same as those of the original samples.

**Tool tip.** A label that appears when you move the mouse pointer over some parts of the PicoScope screen, such as buttons, controls and rulers.

**Trigger.** The part of an oscilloscope that monitors an incoming signal and decides when to begin a capture. Depending on the trigger condition that you set, the scope may trigger when the signal crosses a threshold, or may wait until a more complex condition is satisfied.

**Vertical resolution.** The number of bits that the scope device uses to represent the signal level. This number depends on the design of the device, but can be boosted in some cases by using [resolution enhancement](#).

**View.** A presentation of data from a scope device. A view may be a [scope view](#), an [XY view](#) or a [spectrum view](#).

**Viewport.** The views in the [PicoScope window](#) are arranged in a [grid](#), and each rectangular area in the grid is called a viewport.



## Index

### %

%buffer% variable 92  
 %file% variable 92  
 %time% variable 92

•

.bmp files 40  
 .csv files 40  
 .gif files 40  
 .mask files 94  
 .png files 40  
 .psdata files  
     converting 112, 199  
     saving 40  
 .pskeys files 102  
 .psmaths files 72, 76, 80  
 .psreference files 83  
 .pssettings files 40  
 .txt files 40

### 5

50 Ω DC inputs 116, 198

### A

AC adapter 203  
 AC coupling 205  
 AC power 100  
 Accelerometer inputs 116  
 Access 3  
 Add Measurement dialog 54  
 Adding a measurement 54  
 Address (contact) 4  
 Advance function 76  
 Advanced measurement settings 55  
 Advanced Options toolbar  
     Channel Labels button 115  
     Notes button 115  
     Rulers button 115  
 Advanced triggering 149, 151  
     types 152, 198  
 Alarms 36, 92  
     save on trigger 174  
 Analog intensity 134  
 Analog offset 117  
 AND logical operator 156

Animated GIF 40  
 Arbitrary waveform generator 138, 198  
     editing window 143  
     files 142  
     importing from channel 145  
 Automotive menu 110  
 Auxiliary (AUX) I/O 149, 198  
 Average (statistics) 21  
 AWG 205  
 Axis 12, 17, 19, 205  
     auto-arrange 51  
     horizontal 12, 17, 19  
     offset 164  
     scaling 120, 164  
     vertical 12, 17, 19  
 Axis scaling 117

### B

Bandwidth limiter 117, 198  
 Battery power 100  
 Beep 92  
 Binary files, exporting 44  
 Bit stream 143  
 Bring channel to front 120  
 Buffer Navigation toolbar 136  
 Buffer Overview 37

### C

CAN bus  
     protocol 191  
 Capture count 21  
 Capture modes 9, 10  
 Capture rate preference 100  
 Capture Setup toolbar 130  
 Capture size preference 99  
 Changing device 160  
 Channel 205  
     selecting in a view 51  
 Channel Labels 49, 115  
 Channel Options  
     button 116  
     menu 117  
 Channel ordering 120  
 Channel settings  
     in Properties sheet 30  
 Channel warning symbol 32  
 Channels toolbar  
     PicoLog 1000 Series 125  
     standard 116  
     USB DrDAQ 127

Clipboard 48  
 Close file 39  
 Collection time units 99  
 Color preferences 105  
 Column auto-width 53  
 Command line syntax 199  
 Connect Device dialog 39, 111  
 ConnectDetect 115  
 Contact information 4  
 Converting data files 112, 199  
 Copy  
     as image 48  
     as text 48  
 Copyright 3  
 Coupling control 116  
 Create New Custom Probe dialog 59  
 CSV files, exporting 43  
 Cursors (see Rulers) 23, 24, 28  
 Custom grid layout dialog 52  
 Custom Probe Manager 57  
 Custom Probe wizard 59  
     Create New Custom Probe dialog 59  
     Edit Existing Custom Probe dialog 60  
     Edit Range dialog 66  
     Edit Range dialog (Advanced tab) 68  
     Finished dialog 71  
     Lookup-table Scaling dialog 63  
     Manual Ranges Setup dialog 65  
     Probe ID dialog 70  
     Probe Output Units dialog 61  
     Range Management dialog 64  
     Scaling Method dialog 62  
 Custom probes 31  
     dialog 58  
     saving 39

**D**

Data files  
     converting 112  
 DC coupling 205  
 DC IN power 203  
 DC offset 117, 198  
 Dead time 205  
 Decoding tab 34  
 Default Print Settings 104  
 Delay function 76  
 Demo device 146  
 Demo mode 146, 147, 205  
 Demo Signals  
     dialog 147  
     menu 146

Details dialog  
     channel labels 50  
     customer details 50  
     vehicle details 50  
 Device feature table 198  
 Device, how to change 160  
 Digital color 134  
 Digital inputs 123, 198  
 Digital Inputs button 116, 123  
 Digital outputs 126  
     USB DrDAQ 129  
 Digital Setup dialog 123  
 Digital trigger dialog 155  
 Digital view 15  
     context menu 16  
 DrDAQ 127  
 Dropout trigger 152

**E**

Edge trigger 152  
 Edit Existing Custom Probe dialog 60  
 Edit menu 48  
 Edit Range dialog 66  
     Advanced tab 68  
 Edit Reference Waveform dialog 84  
 Effective resolution 118  
 Email addresses 4  
 Equation dialog 76  
 Equivalent-time sampling 149  
 ETS 149, 205  
     and Advanced Triggering 151  
 Exit 39  
 Exporting data 43  
     binary format 44  
     text format 43  
 External (EXT) trigger 149, 198

**F**

Fall time  
     threshold 55  
 Falling edge button 149  
 Fax number 4  
 Feedback 109  
 File conversion 112  
 File menu 39  
 Filter Method dialog 69  
 Filtering 117  
     channels 121  
     measurements 21  
     statistics 55

Fitness for purpose 3  
 Flexible power 203  
 Flexible Resolution 198  
 FlexRay  
     protocol 196  
 Focus 205  
 Frequency counter 116, 198  
 Frequency difference, measuring 162  
 Frequency legend 24, 28  
 Frequency rulers 24  
 Functions, mathematical 76

## G

General preferences 99  
 Generate mask 94  
 Generate Mask dialog 96  
 Glitches, finding 152  
 Glossary 205  
 Graticule 12, 17, 19, 205  
 Grid 205  
     layout 51, 52  
 Groups, digital input 123

## H

Hand tool 158  
 Harmonic control for measurements 55  
 Help menu 109  
 Horizontal axis 12, 17, 19  
 House current 100  
 Hysteresis 154

## I

I<sup>2</sup>C  
     protocol 190  
 I<sup>2</sup>S  
     protocol 191  
 IEPE inputs 116  
 Image, saving as 40  
 Import from a Channel dialog 145  
 Input impedance 116  
 Interpolation  
     linear 101  
     sin(x)/x 101  
 Interval trigger 151, 152  
 Invert button 32

## K

Keyboard maps 102  
 Keyboard shortcuts 102, 158

## L

Language preferences 103  
 LED  
     on USB DrDAQ 128  
 Legal statement 3  
 Liability 3  
 LIN bus  
     protocol 194  
 Line thicknesses 105  
 Link file 90  
     creating 178  
 Lissajous figures 17  
 Logic trigger 152  
     dialog 156  
 Lookup-Table Scaling dialog 63  
 Lowpass filtering 69, 118, 121, 198

## M

Macro  
     Recorder 97  
     running from command line 199  
 Mains power 100  
 Manual Ranges Setup dialog 65  
 Marquee Zoom tool 158  
 Mask Limit Testing 35, 57  
     How To 171  
 Masks  
     colors 35, 105  
     displaying 51  
     editing 95  
     export 94  
     generate 94  
     import 94  
     in Buffer Overview 37  
     library dialog 94  
     menu 94  
     polygons 95  
     selection dialog 35  
 Math Channel Wizard  
     Equation dialog 76  
     Finished dialog 82  
     Introduction dialog 75  
     Name and Color dialog 80  
     overview 74  
     Units and Range dialog 81  
 Math channels 57, 72  
     Built In 72  
     button 116  
     dialog 72

Math channels 57, 72  
 Library 72  
 Loaded 72  
 overview 32  
 saving 39

MATLAB files  
 exporting 44, 83  
 saving 40

Max (statistics) 21

Maximum Waveforms preference 99

Measurement system  
 selecting 103

Measurements  
 adding 21, 53, 54  
 advanced settings 55  
 capture size 99  
 deleting 21, 53  
 editing 21, 53  
 filtering 21  
 font size 53  
 list of types 180  
 menu 53  
 scope 180  
 spectrum 181  
 statistics 21  
 table 21  
 toolbar 107, 137

Menus 38

Metric measurements 103

Min (statistics) 21

Missing events, finding 152

Mission-critical applications 3

MSO 205  
 setup 123  
 view 14

**N**

NAND logical operator 156

New features 2

NOR logical operator 156

Normal Selection tool 158

Notes 48, 115  
 importing from command line 199

Notes area 48

**O**

Offset 164  
 analog 117

Open file 39

OR logical operator 156

Oscilloscope 7  
 Overrange indicator 12, 116  
 Overvoltage  
 BNC Shell to chassis 13  
 Normal measurement range 13

**P**

Page Down key 39  
 Page Up key 39  
 Panning 159  
 Pattern trigger 155  
 PC Oscilloscope 8  
 Persistence mode 20  
 button 130  
 enabling and disabling 107  
 options 134

Phase rulers 25  
 partition 27  
 units 27  
 wrap 27

PicoLog 1000 Series 125, 126

PicoScope 6 1, 2, 8  
 how to use 3, 6, 7  
 main window 11

Pointer tool tip 22

Polygon 95

Post-trigger delay 185  
 arrow 18  
 control 149, 185

Power Management preferences 100

Power supply 203

Preferences 57  
 capture rate 100  
 colours 105  
 Default Print Settings 104  
 device selection 107  
 dialog 98  
 general 99  
 keyboard 102  
 language 103  
 persistence modes 107  
 power management 100  
 sampling 101  
 spectrum modes 107  
 updates 108

Pre-trigger delay 185  
 control 149, 185

Printing 39  
 from command line 199  
 from menu 39  
 preferences 104

Printing 39  
 preview 39  
 Probe 205  
 custom 31  
 ID dialog 70  
 Output Units dialog 61  
 Progress bar 132  
 Progressive mode 205  
 Properties sheet 30  
 displaying 51  
 Pulse width trigger 151, 152  
 PWM output  
 PicoLog 1000 Series 126  
 USB DrDAQ 129

## R

Range control 116  
 Range Management dialog 64  
 Rapid triggering 149, 198  
 Reference waveforms 57  
 adding 51  
 delay 120  
 dialog 83  
 Library 83  
 Loaded 83  
 overview 33  
 using in equations 76  
 Reset 'Don't show this again' dialogs 99  
 Resolution control 130  
 Resolution enhancement 117, 118, 205  
 Revolutions per minute 28  
 RGB LED on USB DrDAQ 128  
 Rise time  
 threshold 55  
 Rising edge button 149  
 Rotation rulers 25  
 partition 27  
 units 27  
 wrap 27  
 RPM 28, 107  
 RS232/UART  
 protocol 188  
 Rulers 12, 17, 19, 115  
 definition 205  
 deleting 23, 24  
 handles 12, 17, 19  
 legend 28  
 lock button 28  
 phase 25  
 rotation 25  
 settings 27

time 12, 19  
 voltage 12, 17, 19  
 Runt pulse trigger 152, 198

## S

Sampling preferences 101  
 Sampling rate 130  
 Save As 39  
 dialog 40  
 Save file 39  
 Save on trigger 92, 174  
 Scaling 8, 164  
 button 120  
 Scaling Method dialog 62  
 Scope device 205  
 Scope measurements  
 AC volts 180  
 Cycle time 180  
 DC volts 180  
 Duty cycle 180  
 Fall time 180  
 Falling rate 180  
 Frequency 180  
 High pulse width 180  
 Low pulse width 180  
 Maximum 180  
 Minimum 180  
 Peak to peak 180  
 Rise time 180  
 Rising rate 180  
 Scope mode 9  
 button 130  
 Scope view 10, 12  
 Scrolling 164  
 Selection tool, normal 158  
 Send channel to back 120  
 Serial decoding 34, 57  
 data window 90  
 dialog 85  
 link file 90  
 protocols 187  
 Serial number  
 of oscilloscope 109  
 Settings  
 saving 39  
 Signal difference, how to measure 161  
 Signal generator  
 button 138  
 dialog 138  
 sweep mode 138, 198  
 triggering 198

Signal generator  
 USB DrDAQ 141  
 Waveform types 183

Signal rulers 12, 17, 19, 23

Sinx(x)/x preferences 101

Slow sampling transition 101

Smoothing 101

Software version 1

Sound file 92

Space bar 148

Spectrum measurements  
 Amplitude at peak 181  
 Frequency at peak 181  
 Intermodulation Distortion (IMD) 181  
 Signal to Noise Ratio (SNR) 181  
 Spurious-free Dynamic Range (SFDR) 181  
 Total Harmonic Distortion (THD) 181  
 Total Harmonic Distortion plus Noise (THD+N) 181  
 Total power 181

Spectrum mode 9  
 button 130  
 enabling and disabling 107

Spectrum options  
 bins 132  
 dialog 132  
 display mode 132  
 scale 132

Spectrum view 10, 19  
 how to set up 166

SPI  
 protocol 189

Spreadsheet, exporting to 40

Spurious edges, finding 152

Standard deviation 21, 205

Start/Stop toolbar 107, 148

Startup Settings menu 46

State trigger 155

Statistics 21  
 filtering 55

Support 3

Sweep mode 138, 198

Symbols  
 red warning 12  
 yellow warning 32

System requirements 5

Thresholds, digital input 123

Time difference, how to measure 162

Time gate 30

Time rulers 12, 19, 24

Timebase controls 130

Tool tip 205

Toolbars 114

Tools menu 57

Trace 7

Tracking rulers 28

Trade marks 3, 4

Trigger 149, 185, 205  
 advanced 149, 151  
 digital 155  
 dropout 152  
 dual-edge 151  
 edge 152  
 glitches 152  
 interval 151, 152  
 logic 152  
 marker 18  
 missing events 152  
 mode control 149  
 pulse width 151, 152  
 reference point 185  
 runt pulse 152  
 timing 185  
 toolbar 149  
 window 152

Triggering  
 toolbar 107

## U

U.S. measurements 103

UART  
 protocol 188

Undo zoom 158

Updates 109

Upgrades 3

Usage 3

Usage statistics 108

USB DrDAQ 127

USB power 203

'Used' check box 156

User's Guide 109

## T

Telephone number 4

Text files, exporting 40, 43

Threshold for measurements 55

## V

Valid edges, finding 152

Version number  
 hardware 109

Version number  
  software 1, 109  
Vertical axis 12, 17, 19  
Vertical resolution 205  
View 205  
  enabling sub-views 51  
  how to move 163  
  menu 51  
  scope 12  
  selecting channels 51  
  spectrum 19  
  XY 17  
Viewport 205  
Viruses 3

## W

Warning symbol 116  
  red 12  
  yellow 32  
Waveform 7, 12  
  saving 39  
Waveform buffers  
  number of 99  
What's new 2  
Window functions 132, 184  
Window trigger 152

## X

X axis, configuring 51  
X-Axis command 17  
XNOR logical operator 156  
XOR logical operator 156  
XY view 17

## Z

Zero offset 118  
Zooming 164  
  undo 158  
  Zoom overview 159  
  Zooming and Scrolling toolbar 158  
Z-ordering 120





## Pico Technology

James House  
Colmworth Business Park  
ST. NEOTS  
Cambridgeshire  
PE19 8YP  
United Kingdom  
Tel: +44 (0) 1480 396 395  
Fax: +44 (0) 1480 396 296  
[www.picotech.com](http://www.picotech.com)